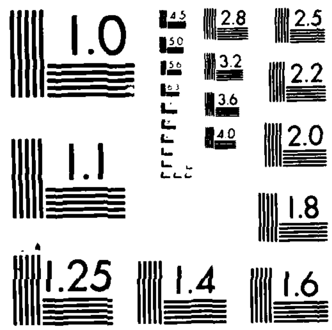
AD-A159 243 TIME DELAY OF ARRIVAL LOCATION ASSESSMENT USING FOUR SATELLITES(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING C M WOZNIAKOWSKI DEC 84 1/2

UNCLASSIFIED AFIT/GSO/PH/84D-5 F/G 17/7 NL



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS 1963-A

TIME DELAY OF ARRIVAL LOCATION

ASSESMENT USING FOUR SATELLITES

THESIS

Chester M. Wozniakowski
Captain, USAF

AFIT/GSO/PH/84D-5

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GSO/PH/84

TIME DELAY OF ARRIVAL LOCATION

ASSESMENT USING FOUR SATELLITES

THESIS

Chester M. Wozniakowski
Captain, USAF

AFIT/GSO/PH/84D-5

AFIT/GSO/PH/84D-5

TIME DELAY OF ARRIVAL LOCATION ASSESSMENT USING FOUR SATELLITES

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Masters of Science in Space Operations

Chester M. Wozniakowski, M.S.
Captain, USAF

December, 1984

PREFACE

Nięch bedzie pochwalony Jezus Chrystus.

Na wiecki wiecki. Amen.

I would like to thank Maj James J. Lange for his aid in selecting this topic (aka "Door #3") and guidance through this project. But especially for his never ending patience in helping me, I want to mention my deepest appreciation. And I want thank LtCol Coleman for assistance in the debugging of the programs, and his never ending supply of patience. And a special thank you to Mr. Joe Marshall, HQ AFTAC, Patrick AFB Fl, for providing the algorithm used to solve the system of four equations used in this thesis.

Also, I would like to express my thanks to all of my fellow classmates who helped me to graduate through the many group study sessions, and their moral support, in keeping with the class motto: "Cooperate and Graduate".

CMW

Table of Contents

List of Figures

List of Tables

Abstract

The Time of Arrival (TOA)/ Time Delay of Arrival (TDOA) concepts for locating a source are reviewed. The Vela satellite program and how they used in conjunction with the TOA concept is discussed. NAVSTAR/GPS is reviewed next and how this interrelates with the other concepts discussed is mentioned. The last concept mentioned is that of nuclear detection. Again, examples on how this idea interrelates with the previous mentioned concepts are shown.

The problem addressed here is to develop mathematical formulas that describe the problem of having four satellites viewing an event, then accurately determining the location and time for that event. The first method uses a TOA technique to solve for the event's location and time. The second method employs the solution from the first method to predict uncertainty in loction and time. Also this uncertainty is determined through the use of implicit differentiation. These results are then compared with the projected difference from the standard solution when the associated value is varied by an equivalent amount, in the first method. Results from one example are discussed.

TIME DELAY OF ARRIVAL LOCATION ASSESSMENT USING FOUR SATELLITES

by

Chester M. Wozniakowski

Capt,USAF

I

Introduction

Background:

The military has had a long desire to know its´ own location, and to navigate to certain locations, reliably. With the space age, this need to navigate and to know one´s own location has increased. Various systems have been developed to meet these requirements. Among the first satellite programs was TRANSIT, a satellite navigation and positioning system for the US Navy in the early 1960´s (4:40). In the early 1970´s, there were two competing concepts, TIMATION (US Navy) and Program 621B (US Air Force), to accomplish the requirements. In 1973, the Defense System Acquisition Review Council approved one navigation system, called Global Positioning System (GPS), which incorporated ideas from both concepts (4:40, 13:46, 21:1177). GPS, also known as NAVSTAR, was initially planned to consist of 24 satellites, but has since been reduced to 18 (5:39, 21:1180). The satellite were to be inclined 60-70 degrees, initially (to be changed to a approximately 55 degree inclination, when the constellation is completed), in 12 hour orbits, at an altitude of approximately 20,000 km (11,000 nm)

(13:46, 18:22, 19:1180-1).

The satellites in the GPS constellation are continuously broadcasting over two different frequencies. Each frequency is an integer multiple of the internal clock. One signal is used for easier acquisition of the satellite's signal, and for crude approximation of one's position. The other signal is designed for specific users and greater accuracy. Also, this signal is encoded which make it difficult to be interfered with, or be accessed by the enemy (4:36).

Depending on the user and the type of receiver, the user is able to determine his position, or velocity. The user determines his position by measuring range and range rate to four broadcasting satellites in his field of view (18:22-3). He determines distances from the satellite by measuring the transmission delay in the satellite's code (4:35). The code contains a best estimate of a satellite's position and drift of its internal clock (4:35-6). With similar information from four satellites, the user has four simultaneous equations with four unknowns to determine his position and time (18:22). Velocities are determined from Doppler processing of the received signal (6:48).

The current set of 11 NAVSTAR satellites, Block 1, will be replaced by a new set of satellites, Block 2. Block 2 satellites will carry aboard secondary payloads, one of which is a nuclear detonation sensor system, called Integrated Operational NUDET (Nuclear Detonation) Detection System (IONDS) (5:38-39). Because of the accurate position and time of the NAVSTAR satellites, GPS is able to give an accurate time and position when the sensors detect

a nuclear detonation. By means of a sensor crosslink system, a satellite is able to pass on the information of the detected nuclear explosions and its assessment of nuclear attacks to another satellite and then down to the ground station (5:89, 19:1131).

A constellation of satellites with a nuclear detection sensor system might be able to reduce some of the uncertainties in the treaty verification process. An example of a situation where such a system could have been proven fruitful occurred in September, 1979 near Antarctica. Vela satellites, first launched in 1963, have a mission "to detect atmospheric nuclear explosions" (12:69); i. e. treaty verification of the Nuclear Test Ban Treaty. One satellite detected a possible nuclear detonation on the night of September 22, 1979, that indicated a possible treaty violation (12:67). The possible detonation was too small for the Vela detonation locator sensor to determine the location of the event. Only one satellite with nuclear detection capability observed the possible event. Given this fact and the Vela satellites orbit, the only land area within this Vela's coverage was South Africa. This incident could also be interpreted as "one of the 'zoo' events (unexplained anomalous signals obtained from Vela satellites), possibly a consequence of the impact of a small meteoroids on the satellite" (12:67). As NAVSTAR offers continuous worldwide coverage, with a minimum of four satellites in a field of view, and is a good locator, it offers the possibility of eliminating some of the uncertainties of treaty verification.

Statement of Problem:

We have seen that an army solider with a back pack can find his own location and time when at least four satellites are in the soldier's field of view. This is also true for planes, or satellites.

And we have seen that a nuclear detection device, such as IONDS, on board the NAVSTAR GPS constellation of satellites offers the potential to accurately determine the position and time for a nuclear explosion.

The problem that shall be addressed is to develop mathematical formulas that describe the problem of having four satellites viewing an event, then accurately determining the location and time of that event. This will first be done for an ideal case. Then, the problem will be look at when non-ideal conditions exist, such as uncertainties due to satellite's position or atmospheric transmission delays.

The following assumptions are made in this paper:

1. The sensor that will be used is a visual (optical) sensor. The paper is not concern with the mechanics of how the sensor works or with other type of sensors.

2. There is only a single event. Multiple events are beyond the scope of this paper.

3. Only four satellites see the event. Less than four satellites yield an under determined set of simultaneous equations, and thus unable to determine the location (X,Y,Z) and time uniquely. If more than four satellites observe the event, then the solution set is over determined. Through statistical analysis, one

to the ith receiver arriving at time ti. Multiplying out the equation, one gets

$$xs^2 - 2xsx1 + x1^2 + ys^2 - 2ysy1 + y1^2 + zs^2 - 2zsz1 + z1^2 = d1^2 - 2dsd1 + ds^2 \quad (2a)$$

$$xs^2 - 2xsx2 + x2^2 + ys^2 - 2ysy2 + y2^2 + zs^2 - 2zsz2 + z2^2 = d2^2 - 2dsd2 + ds^2 \quad (2b)$$

$$xs^2 - 2xsx3 + x3^2 + ys^2 - 2ysy3 + y3^2 + zs^2 - 2zsz3 + z3^2 = d3^2 - 2dsd3 + ds^2 \quad (2c)$$

$$xs^2 - 2xsx4 + x4^2 + ys^2 - 2ysy4 + y4^2 + zs^2 - 2zsz4 + z4^2 = d4^2 - 2dsd4 + ds^2 . \quad (2d)$$

In order to linearize the above equations with respect to the variables (xs,ys,zs,ds), one can use one equation as a base line and then find the difference between that one and the other equations. Let equation (2a) be the base line equation, then equation (3a) is defined to be the difference between (2a) and (2b). Similarly, the other equations can be defined. Thus one gets the following equations.

III

Formula Derivation

As stated earlier, we shall look at the problem of developing mathematical formulas to describe four satellites viewing some event, and determining the location and time of that event, following the procedures outlined by Marshall (17). Then, we shall look at what errors occurred due to the uncertainties in position of the satellite, or the time aboard the satellite. This will be done by the use of the commonly used propagation of error formula.

General Solution:

Let (xs,ys,zs,cts) define the source of the event that has been observed by n sensors. Also, let (xi,yi,zi,cti) define the location of the ith satellite sensor in a cartesian coordinate system. The distance ct is that distance travelled by light in time interval t where c is the speed of light. For convenience, let cti = di and cts = ds.

Then the formulas that define an event as viewed by n sensors can be written as follows:

$$(xs-x1)^2 + (ys-y1)^2 + (zs-z1)^2 = (ds-d1)^2 \qquad (1a)$$
$$(xs-x2)^2 + (ys-y2)^2 + (zs-z2)^2 = (ds-d2)^2 \qquad (1b)$$
$$(xs-x3)^2 + (ys-y3)^2 + (zs-z3)^2 = (ds-d3)^2 \qquad (1c)$$
$$(xs-x4)^2 + (ys-y4)^2 + (zs-z4)^2 = (ds-d4)^2 \qquad (1d)$$

where the term on the right hand side of each equation represents the distance travelled by light in going from the source at time ts

reflecting off meteoroids or particles near the sensor, or the impact of small meteorites on the sensor (12:71-72).

As stated earlier, the NAVSTAR GPS satellite constellation will generally offer complete earth coverage with at least four satellites. As noted before, these satellites will carry a secondary package to detect clandestine nuclear explosions and to assess nuclear attacks (7:220, 24:6). Thus, the NAVSTAR system has the potential to be a self-corroborating system, that is, more than one satellite will see the event. Also, because of the accuracy of the clocks on board, and the satellite ephemeris, GPS offers the potential to accurately determine the location and time of such nuclear explosions. And the rapid rise of the first pulse of the optical nuclear signal allows easy time-tagging, that is, determining exact time of arrival of the signal at a detector.

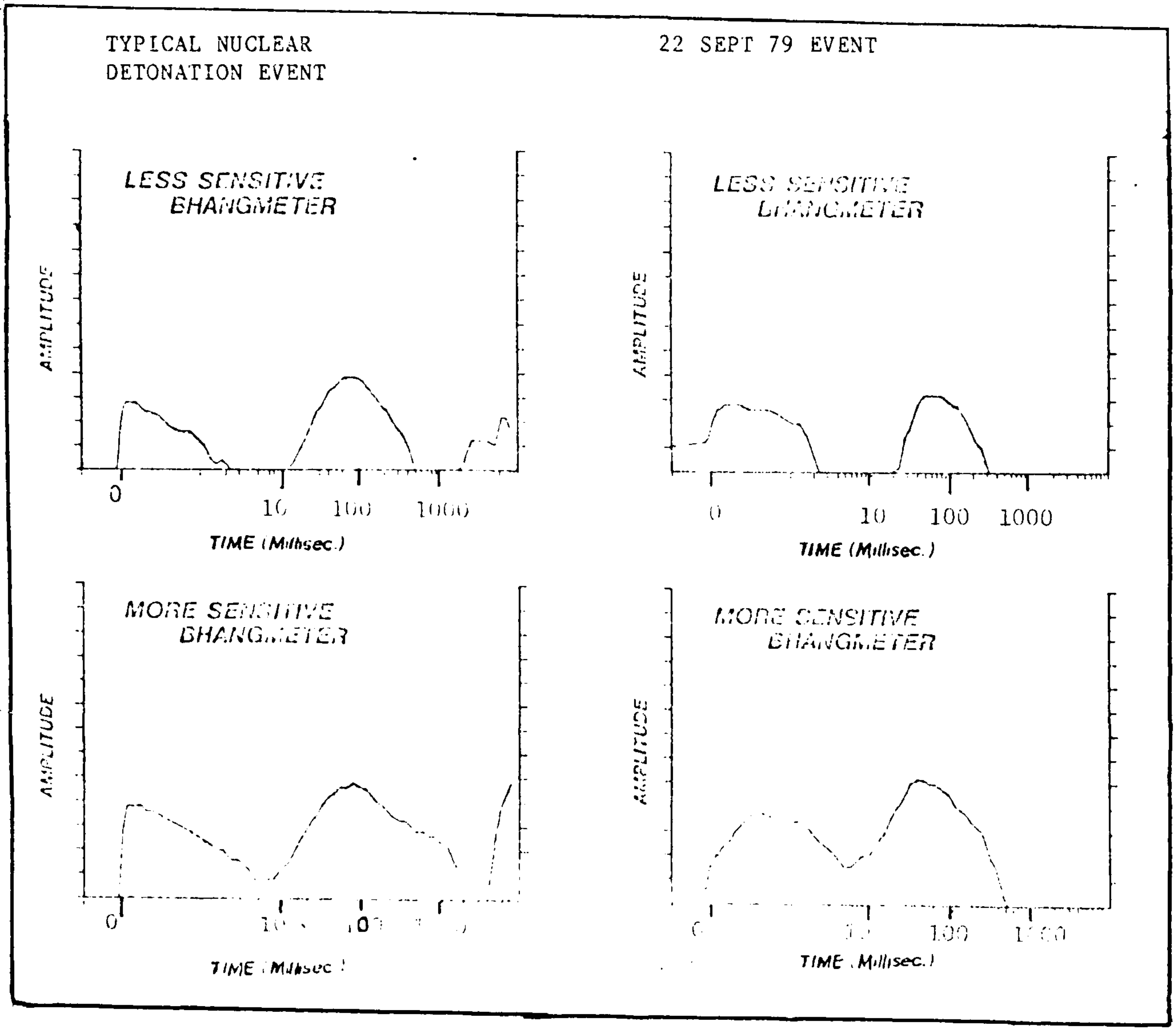


Figure 3: Example of Bhangmeter Signals (12:67).

was initially thought to be a clandestine nuclear test. Lack of corroborating evidence from other ground detection sensors and other space sources (10, 12:71-72) was one indication that this event was not a nuclear explosion. A panel of experts came to the conclusion that the received signal "was one of the zoo events [unexplained anomalous signals obtained from Vela satellites]" (12:67). Zoo events are thought to be caused by sunlight glints

45).

Later, Glasstone mentions various methods for detection of nuclear blasts. Among the ideas mentioned is that of airborne radioactive debris detection. Even though this gives a good positive indicator that a nuclear explosion has a occurred, it is a poor indicator on the source of the explosion due in part to the meteorological conditions (e.g. winds) (8:683). On the other hand, the use of seismic waves or acoustic waves allows for a good determination of the source or location of the explosion. But these are not necessarily good indicators of the event happening. For example, seismic waves from an earthquake might be confused as an underground nuclear detonation (8:686-7). Glasstone also explains how satellites might be used to detect nuclear explosions in space. Sensors would detect X-rays, gamma rays, neutrons, and electrons from the burst (8:695-7).

The main mission of Vela satellites is "to view the earth, and to detect atmosphere nuclear explosions." (12:69) Each Vela satellite carries two similar sensors, called Bhangmeter, to perform this mission by sensing nuclear explosions′ very brief intense bursts of light. Lighting and cosmic rays can be differentiated from nuclear explosions. Nuclear explosions generate two distinct pulses in a very short period of time, whereas lighting produces only one pulse, and cosmic rays affect only one of the twin sensors (12:69). (See Figures 1 and 2 for a comparison.)

From Figure 3, one can see why when on September 22, 1979, one Vela satellite saw an event off the coast of Antarctica, that it

blast.

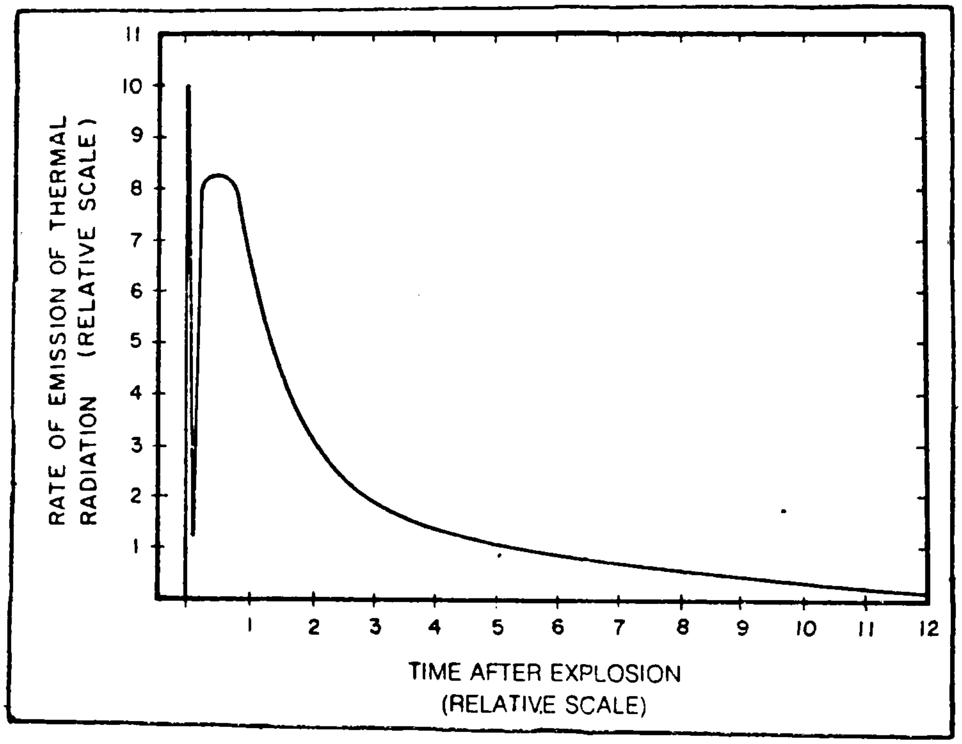


Figure 2: Emission of thermal radiation in two pulses in an air blast (8:45).

The first pulse lasts about a tenth of a second for a 1 megaton burst. Most of the radiation is located in the ultraviolet region of the spectrum, due to the high temperatures involved. However, the second pulse may last for several seconds. Also, the temperatures are significantly lower than in the first pulse. Thus, the radiation is now located in the visible and infrared sections of the spectrum. It is this latter radiation which accounts for most of the radiation in a nuclear explosions (8:44-

concept, the user is able to determine his position and time from four satellites (6:48). The geometry of the satellites with respect to the receivers (users) has an impact on the users position errors. The effect of the geometry is expressed by the geometric dilution of precision (GDOP) parameters. The value of GDOP itself is a composite measure that reflects the influence of satellite geometry on the combined accuracy of the estimate of user time (user clock offset) and user position. The four 'best' satellites selected by the user receivers are those with the lowest GDOP (18:10). The velocity of a vehicle is found in a similar manner, except the velocity is determined from Doppler processing of the received signal (6:48).

Among the obvious usages of such a system are the location finding for a vessel, and the velocity that it is traveling at. Another is to travel from point A to point B (6). Future planned usage includes determining ocean currents within 1 to 2 cm over a 5 minute averaging interval (20:31). Another planned usage for the NAVSTAR system is to update inertial guidance systems and to improve the accuracy of mobile or air-launched strategic missiles (13:47). Also, GPS is planned to be part of future LANDSAT spacecrafts and navigation system for the Space Shuttle (7:222).

Nuclear Detection:

Samuel Glasstone, in his authoritative book, Effects of Nuclear Weapons, describes how a nuclear bomb works and what are the effects from the nuclear explosions. In his description of air and surface nuclear bursts, he discusses the thermal radiation from an air blast. There are two pulses of thermal radiation from the

NAVSTAR:

NAVSTAR will consist of 18 satellites with three on-orbit spares in six orbital planes, equally spaced 60^{o} apart. Each orbital plane will contain three satellites, 120^{o} apart. The satellites will have an inclination of 55^{o} (21:1180).

The NAVSTAR satellites will broadcast their position and time on two broadcasting frequencies, each a multiple of the clock oscillation 10.23 MHz. The maximum allowable uncertainties for the clock is one part per 10^{12} per day (19:5). The first broadcasting signal, L_1, also known as C/A signal for coarse/acquisition, propagates at 1575.42 MHz. The signal is unique to each satellite and consist of pseudo-random noise chip stream which repeats every millisecond (15:1196, 19:6). The purpose of this signal is to allow for easy acquisition of the satellite's signal and a coarse estimation of one's position. The other broadcasting signal, L_2, also known as the P code for precision, propagates at 1227.6 MHz. The signal consist of a 7 day-long phase of a complete 267 day cycle, which makes the signal more jam resistant and permits access to only friendly forces (14:1196, 18:6). The concept allows for an easy acquisition of a satellite signal by means of the C/A signal, then to transfer over to the P code to get a more precise fix on one's location. Also, the C/A would be available to the general public, e.g. commercial aviation, while the P code would be restricted for military usage (21:1182).

As the satellite signal identifies each satellite uniquely, a user locates that satellite in a Earth-center, Earth-fixed coordinate system and establishes the system time. Using the TOA

image in the orbital plane (14:L86, 22:2). A fourth satellite should yield a single point as the source (22:2). Using Vela satellites and other deep space sensors, such as Prognoz and Uhuru, in conjunction with other investigative tools, scientists were able to discern that these sources are outside of the solar system (22:3, 26:10-12).

This principle has been used also when looking at the earth. Turman reported how Vela's optical sensors assisted in the detection of lighting "superbolts". These superbolts radiate energy in the order of $10^{11} - 10^{13}$ Watts within 1 millisecond, approximately 100 times stronger than normal (see Figure 1) (30:2566).

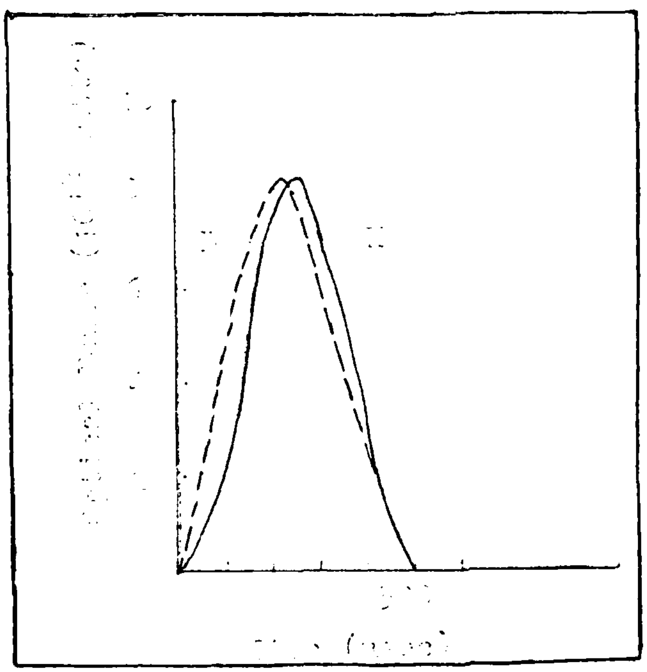


Figure 1: Superbolt pulse shape
H: high-sensitivity detector
L: low-sensitivity detector (30:2567).

Vela:

The Vela satellites were first launched in pairs in October, 1963 (12:69). Vela satellites were launched into approximately 30 degree inclinations, near circular orbits at altitude of 60,000-70,000 mi. (approximately 13 earth radii, or 120,000 km) and a period of 112 hours (2:L157, 9:3866-3867, 12:69, 28:279). The satellite pairs were about 180 degrees apart from each other (2:L157, 28:279). Each satellite rotated about its spin axis approximately every 64 seconds, actively maintained in the direction of earth (2:L157, 28:279).

However, each X-ray collimator sensor is attached perpendicular to the spin axis. Thus, an angular strip between 11^{o} and 12^{o} relative to the spin axis is viewed every 64 sec, or 1 spin. And as the satellite orbits around the earth, always pointing toward earth, the celestial sphere is observed once every half orbit, or 56 hrs (2:L157, 9:279). Gamma and X-ray events in the heavens have been observed and recorded by satellites.

In one example by Connors and others (2), they used the data from Vela satellites to detect a new X-ray source in the southern sky. Still, in another example by Terrell and others (28), two gamma ray bursts were discovered by Vela.

The spin to the Vela spacecraft causes the X-ray sensor (with a small conical field of view) to trace out a circle on the celestial sphere (30:5). From the general principle of TOA, two spacecraft seeing an event define a circle, which encompasses the event. Three spacecraft define the intersecting circles, whose points of intersection represent the source position and its mirror

other techniques in conjunction with the TOA concept, they discovered that these sources were located in the same spiral branch of the galaxy as the Sun, or in another galaxy. In one case, they located the exact location to within 4-5 degrees of arc (22:2).

Proctor in his studies of lighting in South Africa was the first to determine the location of sferics (atmospheric interference) in a three dimensional fix. He was able to determine the location by tracking the time differences between the time at which the pulse was received at four VHF receivers (23:1478).

Uman and others described a study of a three-stroke lighting bolt that struck a weather tower at Kennedy Space Center. They used the TOA technique to locate lighting channels inside thunderstorms, which consisted of the measured time delay between a flash and the arrival of a corresponding sound of thunder at the 25 station network (30:11).

Rustan and others continued the study of lighting bolt at Kennedy Space Center. Employing the Kennedy Space Center Lighting Detection and Ranging system, they were able to determine the three dimensional location by measuring the difference in the time of arrival of radiated pulse at four ground stations, and calculating the locations from the data (25:4893)

Toman and Martine (29) discussed the usage of the TOA concept in conjunction with locating the origin of natural or man-made seismological disturbances. Also, Wood and Treitel described how time differences between reflected signals can represent structural deformation, which aids in oil exploration (33:649).

of the earth. Thus with three receivers, a location in three dimensions (X,Y,Z) can be determined. However, to determine an associated time for the event, a fourth sensor is needed. From linear algebra, this problem is described by four equations and four unknowns, which yields a unique solution. When one equation is missing, the system of equations is under developed and does not yield a unique solution. When there are five or more equations, the system is over developed. However, through statistical analysis, a solution with the smallest uncertainty is found.

Time Difference of Arrival (TDOA) is very similar to the TOA concept. However, now the key measurement is the difference in arrival of the signal as compared with some baseline reception. Even with these distinctions, the terms TOA and TDOA are used interchangeably in the literature.

The TOA concept was used by LaBahn and Paul (16) to gain a better understanding of ionospheric height variations, particularly E and F regions, through 5 and 15 MHz signals. An experiment was established to accurately measure the time of arrival of precisely controlled HF transmitters over a fixed generally one-hopped, mid-latitude path.

Also, in a study of Beacon Tracking System, Colquitt (3) used a simulation that consisted of generating synthetic TDOA data for a given test set up, to test such a system.

The TOA/TDOA concept was not restricted to the Earth. Prilutskiy, Rozenthal and Usov used the TOA concept in determining that the source of gamma ray bursts was not the Sun, the Moon or the Earth. Also, they were able "to determine that the sources were not necessarily located in the galactic plane" (22:2). Using

II

Literature Review

The Time of Arrival (TOA)/ Time Delay of Arrival (TDOA) concept and some of their usages are looked at first. The second idea that is discussed is the Vela satellite system, and how it has been employed with respect to the TOA/TDOA concept. The next topic to be discussed is the NAVSTAR (GPS) navigation system and how it relates to the TOA/TDOA concept. Again, examples on how NAVSTAR could be used will follow. The concept of nuclear detection are also examined, and how it relates to the TOA/TDOA concept are shown. Then these four ideas are merged to formulate the problem.

Time Of Arrival Concept:

Time of arrival (TOA) is a type of measurement used to measure the amount of time (how long) it takes a signal to reach a receiver from an emitter. Knowing how fast the signal travels and the position of the receiver, one is able to determine how far away the signal is from you. Using a sufficient amount of receivers, then one can pinpoint the emitter's location at a given time.

The reverse is also true; that is, one is able to determine one's own position through the use of multiple emitters at known positions.

In the TOA concept, a single receiver limits the location of the emitter to a sphere, two receivers to a circle, and three receivers to two points. Usually one is able to throw one of the points away as being unreasonable, such as being below the surface

is able to determine a unique solution with the least amount of uncertainty However, this again is beyond the scope of this paper.

Methodology:

The TOA problem where four satellites see the same event will be solved first. Using the same technique, the position of the sensors will be varied and we will see how this affects the solution.

The uncertainties of position using the propagation of error technique will be solved for next. The results will then be compared with the solutions from the previous methods.

In the Literature Review section, the main concepts of TOA, Vela, nuclear detection, and NAVSTAR will be discussed first. Then, the mathematical formulas required to solve the four satellite triangulation, and used in the propagation of errors will be derived. Examples on their usage are included. In the next section, the results of the computer runs will be discussed.

$$- 2xsx1 + 2xsx2 + x1^2 - x2^2 - 2ysy1 + 2ysy2 + y1^2 - y2^2 -$$
$$2zsz1 + 2zsz2 + z1^2 - z2^2 = d1^2 - d2^2 - 2dsd1 + 2dsd2 \qquad (3a)$$

$$- 2xsx1 + 2xsx3 + x1^2 - x3^2 - 2ysy1 + 2ysy3 + y1^2 - y3^2 -$$
$$2zsz1 + 2zsz3 + z1^2 - z3^2 = d1^2 - d3^2 - 2dsd1 + 2dsd3 \qquad (3b)$$

$$- 2xsx1 + 2xsx4 + x1^2 - x4^2 - 2ysy1 + 2ysy4 + y1^2 - y4^2 -$$
$$2zsz1 + 2zsz4 + z1^2 - z4^2 = d1^2 - d4^2 - 2dsd1 + 2dsd4 \qquad (3c)$$

Next, one can recombine the terms, transfer all terms containing the variables xs, ys, and zs to the left side of the equation, and transfer all others terms involving constants to the right side. Then one gets

$$2xs(x2-x1) + 2ys(y2-y1) + 2zs(z2-z1) = d1^2 - d2^2 - 2dsd1 +$$
$$2dsd2 + x2^2 - x1^2 + y2^2 - y1^2 + z2^2 - z1^2 \qquad (4a)$$

$$2xs(x3-x1) + 2ys(y3-y1) + 2zs(z3-z1) = d1^2 - d3^2 - 2dsd1 +$$
$$2dsd3 + x3^2 - x1^2 + y3^2 - y1^2 + z3^2 - z1^2 \qquad (4b)$$

$$2xs(x4-x1) + 2ys(y4-y1) + 2zs(z4-z1) = d1^2 - d4^2 - 2dsd1 +$$
$$2dsd4 + x4^2 - x1^2 + y4^2 - y1^2 + z4^2 - z1^2 . \qquad (4c)$$

One could rewrite these equations in matrix form to get

$$\overline{AX} = \overline{BD} + \overline{C} \qquad (5)$$

where

$$\overline{A} = 2 \begin{bmatrix} x2-x1 & y2-y1 & z2-z1 \\ x3-x1 & y3-y1 & z3-z1 \\ x4-x1 & y4-y1 & z4-z1 \end{bmatrix} ,$$

$$\overline{X} = \begin{bmatrix} xs \\ ys \\ zs \end{bmatrix}, \qquad \overline{B} = \begin{bmatrix} 2d2 - 2d1 \\ 2d3 - 2d1 \\ 2d4 - 2d1 \end{bmatrix},$$

$\overline{D} = [ds]$, and

$$\overline{C} = \begin{bmatrix} d1^2 - d2^2 + x2^2 - x1^2 + y2^2 - y1^2 + z2^2 - z1^2 \\ d1^2 - d3^2 + x3^2 - x1^2 + y3^2 - y1^2 + z3^2 - z1^2 \\ d1^2 - d4^2 + x4^2 - x1^2 + y4^2 - y1^2 + z4^2 - z1^2 \end{bmatrix}. \quad (6)$$

Solving for $\overline{X}$, then

$$\overline{X} = \overline{A}^{-1}\overline{BD} + \overline{A}^{-1}\overline{C} \qquad (7)$$

provided $\overline{A}^{-1}$ exists.

Assuming $\overline{A}^{-1}$ exists, then one can rewrite equation (7) as

$$\overline{X} = \overline{FD} + \overline{H} \qquad (8)$$

where $\overline{F} = \overline{A}^{-1}\overline{B}$ and $\overline{H} = \overline{A}^{-1}\overline{C}$.

But what is $\overline{X}$? $\overline{X}$ is the vector defining the position of the event, $[xs\ ys\ zs]^T$. Thus equation (8) can be rewritten as

$$\overline{X} = \begin{bmatrix} xs \\ ys \\ zs \end{bmatrix} = \begin{bmatrix} f1 \\ f2 \\ f3 \end{bmatrix} [ds] + \begin{bmatrix} h1 \\ h2 \\ h3 \end{bmatrix}. \qquad (9)$$

The three components of $\overline{X}$ are solved in terms of a fourth; that is, xs, ys, and zs can be considered as dependent variables of ds, or the time of the event. The next step is to solved for ds, and thus determine a unique solution.

If one substitutes these values for xs, ys, zs in

equations (1a-1d), one then gets, for example:

$$((f1ds + h1) - x1)^2 + ((f2ds + h2) - y1)^2 +$$
$$((f3ds + h3) - z1)^2 = (d1 - ds)^2 . \qquad (10)$$

Multiplying equation (10) out,

$$(f1ds + h1)^2 - 2(f1ds + h1)x1 + x1^2 + (f2ds + h2)^2 -$$
$$2f2ds + h2)y1 + y1^2 + (f3ds + h3)^2 - 2(f3ds + h3)z1 + z1^2$$
$$= d1^2 - 2dsd1 + ds^2 . \qquad (11)$$

Multiplying equation (11) out still further, one gets

$$f1^2 ds^2 + 2fidsh1 + h1^2 - 2(f1ds + h1)x1 + x1^2 + f2^2 ds^2 +$$
$$2f2dsh2 + h2^2 - 2(f2ds + h2)y1 + y1^2 + f3^2 ds^2 + 2f3dsh3 +$$
$$h3^2 - 2(f3ds - h3)z1 + z1^2 = d1^2 - 2dsd1 + ds^2 . \qquad (12)$$

Rearranging equation (12),

$$f1^2 ds^2 + f2^2 ds^2 + f3^2 ds^2 - ds^2 + 2f1dsh1 + 2f2dsh2 + 2f3dsh3$$
$$- 2f1dsx1 - 2f2dsy1 - 2f4dsz1 + 2d1ds + h1^2 + h2^2 + h3^2$$
$$- 2h1x1 - 2h2y1 - 2h3z1 + x1^2 + y1^2 + z1^2 - d1^2 = 0. \qquad (13)$$

If one lets $\bar{K} = [x1\ y1\ z1]$, then $\bar{K}^T = [x1\ y1\ z1]^T$ and $\overline{KK}^T = [x1^2 + y1^2 + z1^2]$. Then equation (13) can be rewritten in matrix form as

$$(\bar{F}^T\bar{F} - 1)ds^2 + 2(\bar{F}^T\bar{H} - \bar{F}^T\bar{K}^T + d1)ds +$$
$$(\bar{H}^T\bar{H} + \overline{KK}^T - 2\overline{KH} - d1^2) = 0. \qquad (14)$$

But this is nothing more than a quadratic equation, which can be solved. Let $(\bar{F}^T\bar{F} - 1) = L$, $2(\bar{F}^T\bar{H} - \bar{F}^T\bar{K}^T + d1) = M$, and

$(\overline{H}^T\overline{H} + \overline{KK}^T - 2\overline{KH} - d1^2) = N$. Then solving for ds, one gets

$$ds = (-M \pm SQRT\ (M**2 - 4LN))/(2L). \qquad (15)$$

As expected, there are two solutions. Generally, one of these is totally unrealistic and can be disregarded. The selection is based on the real world situation and the experience of the individual.

Once one has a solution for ds, then one can go back and solve for [xs, ys, zs] in equation (5).

Example 1:

An example now would be appropriate to illustrate what has just been said. Let us look at a simplified example where the sensors are equal distance from the event, and the event is at the center of the cartesian coordinate system. Let (x1,y1) equal (0,1.414), (x2,y2) equal (-1,-1), and (x3,y3) equal (1,-1) and d1, d2 and d3 equal 2. When one substitutes these values into equations (1a-1d), one gets the following

$$(xs-0)^2 + (ys-1.414)^2 = (2-ds)^2 \qquad (16a)$$
$$(xs+1)^2 + (ys+1)^2 = (2-ds)^2 \qquad (16b)$$
$$(xs-1)^2 + (ys+1)^2 = (2-ds)^2 . \qquad (16c)$$

Multiplying out, subtracting equations, and recombining terms as in equations (2) through (4), or going directly to equation (5), then

$$\bar{A} = 2\begin{bmatrix} -1-0 & -1-1.414 \\ 1-0 & -1-1.414 \end{bmatrix} = 2\begin{bmatrix} -1 & -2.414 \\ 1 & -2.414 \end{bmatrix}$$

$$= \begin{bmatrix} -2 & -4.828 \\ 2 & -4.828 \end{bmatrix} . \qquad (17)$$

Also, $\bar{B} = 2\begin{bmatrix} 2-2 \\ 2-2 \end{bmatrix} = 2\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} . \qquad (18)$

Likewise, one can solve for the $\bar{C}$ matrix.

$$\bar{C} = \begin{bmatrix} 4-4+1-0+1-2 \\ 4-4+1-0+1-2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} . \qquad (19)$$

Now, solving for $\bar{A}^{-1}$, one gets

$$\bar{A}^{-1} = \frac{1}{19.312} \begin{bmatrix} -4.828 & -4.828 \\ -2 & 2 \end{bmatrix} . \qquad (20)$$

The next step is to solve for $\bar{F}$ and $\bar{H}$ as in equation (7).

$$\bar{F} = \bar{A}^{-1}\bar{B} = \frac{1}{19.312} \begin{bmatrix} -4.828 & -4.828 \\ -2 & 2 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad (21)$$

and

$$\overline{H} = \overline{A}^{-1}\overline{C} = \frac{1}{19.312}\begin{bmatrix} -4.828 & -4.328 \\ -2 & 2 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ 0 \end{bmatrix} . \tag{22}$$

Thus, equation (8) now looks like

$$\begin{bmatrix} xs \\ ys \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} ds + \begin{bmatrix} 0 \\ 0 \end{bmatrix} . \tag{23}$$

If one lets $\overline{K} = [\ 0 \quad 1.414\ ]$, then when one substitutes these values into equation (14) one gets $\overline{F}^T\overline{F} = 0$, $\overline{F}^T\overline{H} = 0$, $\overline{F}^T\overline{K}^T = 0$, $\overline{H}^T\overline{H} = 0$, $\overline{KK}^T = 2$, and $\overline{KH} = 0$. Thus, $L = 0 - 1 = -1$,

$M = 2(0 - 0 + 2) = 2(2) = 4$, and

$N = (0 + 2 - 2(0) - 4) = 2 - 4 = -2$. Then solving for ds by means of the quadratic equation, one gets

$$\begin{aligned} ds &= (-4 \pm \text{SQRT}\ (16 - 4(-1)(-2)))/(2(-1)) \\ &= (-4 \pm \text{SQRT}\ (8))/\ -2 \\ &= 2 \mp \text{SQRT}\ (2) \\ &= .586,\ 3.1414 \quad . \end{aligned} \tag{24}$$

Therefore, when one substitutes these values for ds in equation (23), one can solve for xs, and ys

$$\begin{bmatrix} xs \\ ys \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} .586 + \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{25a}$$

and $$\begin{bmatrix} xs \\ ys \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} 3.414 + \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} . \tag{25b}$$

Error Analysis:

Let xs be defined by some function of variables x1, y1, z1, d1, x2, y2, z2, d2, x3, y3, z3, d3, x4, y4, z4, and d4. In mathematical symbolism, this becomes

$$xs = f(x1,x2,x3,x4,y1,y2,y3,y4,z1,z2,z3,z4,d1,d2,d3,d4). \quad (26)$$

Likewise ys, zs, and ds can be written as some function of the same variables, respectively.

The total differential yields, for example

$$\begin{aligned} dxs = {} & \frac{\partial xs}{\partial x1}(dx1) + \frac{\partial xs}{\partial x2}(dx2) + \frac{\partial xs}{\partial x3}(dx3) + \frac{\partial xs}{\partial x4}(dx4) + \\ & \frac{\partial xs}{\partial y1}(dy1) + \frac{\partial xs}{\partial y2}(dy2) + \frac{\partial xs}{\partial y3}(dy3) + \frac{\partial xs}{\partial y4}(dy4) + \\ & \frac{\partial xs}{\partial z1}(dz1) + \frac{\partial xs}{\partial z2}(dz2) + \frac{\partial xs}{\partial z3}(dz3) + \frac{\partial xs}{\partial z4}(dz4) + \\ & \frac{\partial xs}{\partial d1}(dd1) + \frac{\partial xs}{\partial d2}(dd2) + \frac{\partial xs}{\partial d3}(dd3) + \frac{\partial xs}{\partial d4}(dd4) \quad (27) \end{aligned}$$

where the differential dx1 can be approximated by $\Delta x1$, where $\Delta x1$ is extremely small. Similarly, the other differentials can be approximated. Thus, the above equation can be rewritten as

$$\Delta xs = \frac{\partial xs}{\partial x1}(\Delta x1) + \ldots + \frac{\partial xs}{\partial y1}(\Delta y1) + \ldots + \frac{\partial xs}{\partial d4}(\Delta d4). \quad (28)$$

If the uncertainties in the variables are random in nature, that is independent of each other (1:61-64), then the uncertainty in xs because of uncertainties in each of the variables is given by the expression,

$$\Delta xs = \left[\frac{\partial x}{\partial x1}^2 (\Delta x1)^2 + \ldots + \frac{\partial xs}{\partial y1}^2 (\Delta y1)^2 + \ldots + \frac{\partial xs}{\partial d4}^2 (\Delta d4)^2 \right]^{1/2} . \quad (29)$$

But how does one find the $\partial xs/\partial x1$? This is done by implicit differentiation and the use of Jacobian matrix

$$\frac{\partial xs}{\partial x1} = - \frac{\partial(F,G,H,J)/\partial(x1,ys,zs,ds)}{\partial(F,G,H,J)/\partial(xs,ys,zs,ds)} \quad (30)$$

$\partial(F,G,H,J)/\partial(xs,ys,zs,ds)$ is nothing more than the determinant which consists of the partial derivatives of the functions F, G, H, and J with respect to xs, ys, zs and ds. So the first row is made up of the partial derivatives of F with respect to xs, ys, zs, and ds. $\partial(F,G,H,J)/\partial(x1,ys,zs,ds)$ is again a determinant. However, the first column is replaced by the partial derivatives of F, G, H, and J with respect to x1. If one slightly rewrites (2a-2d) in terms of F, G, H, and J respectively, then one can solve for the partial derivatives of F, G, H, and J with respect to xs, ys, zs, ds, x1, and the other variables. For example,

$$F = xs^2 - 2xsx1 + x1^2 + ys^2 - 2ysy1 + y1^2 + zs^2 - 2zsz1 + z1^2 - d1^2 + 2dsd1 - ds^2 = 0 . \quad (31)$$

Then taking the respective partial derivatives, one gets

$\partial F/\partial xs = 2xs - 2x1$ $\quad \partial F/\partial x1 = -2xs + 2x1$ $\quad \partial F/\partial ys = 2ys - 2y1$

$\partial F/\partial y1 = -2ys + 2y1$ $\quad \partial F/\partial zs = 2zs - 2z1$ $\quad \partial F/\partial z1 = -2zs + 2z1$

$\partial F/\partial ds = 2ds - 2d1$ $\quad \partial F/\partial d1 = -2ds + 2d1$

$$\frac{\partial F}{\partial x2} = \frac{\partial F}{\partial x3} = \frac{\partial F}{\partial x4} = \frac{\partial F}{\partial y3} = \ldots = \frac{\partial F}{\partial d4} = 0. \quad (32)$$

Likewise, one can solve for the respective partial derivatives of

G, H, and J. Next, one can substitute numerical values for the variables, and then substitute those values in the Jacobian determinants (27:187-191).

Example 2:

From the previous example, ti was found that (xs,ys) equalled (0,0) and that ds equalled 3.414 or .586. Thus, when one substitutes the previous values and the new found values into equation (32), one can solve for the determinants and the propagation of error. For example, the denominator determinant (DD) equals

$$DD = \begin{vmatrix} 2(0-0) & 2(0-1.414) & 2(2-.586) \\ 2(0-(-1)) & 2(0-(-1)) & 2(2-.586) \\ 2(0-1) & 2(0-(-1)) & 2(2-.586) \end{vmatrix}$$

$$= \begin{vmatrix} 0 & -2.828 & 2.828 \\ 2 & 2 & 2.828 \\ -2 & 2 & 2.828 \end{vmatrix} = 0[0] - 2[-13.654] + (-2)[-13.654]$$

$$= 54.614. \qquad (33)$$

If on assumes that the only error is in the x-direction and the $\Delta x1$, $\Delta x2$, and $\Delta x3$ equals -.001, .001, and .01, respectively, then one can solve for the respective partial derivatives. For example,

$$\frac{\partial F}{\partial x2} = \frac{1}{DD} \begin{vmatrix} 0 & -2.828 & 2.828 \\ -2 & 2 & 2.828 \\ 0 & 2 & 2.828 \end{vmatrix}$$

$$= 2(-13.654)/\ 54.614 = -.5. \qquad (34)$$

Similarly, the other components can be derived so that

$$xs = [0(-.001)^2 + (-.5)^2(.001)^2 + (-.5)^2(.01)^2 + 0 + \ldots + 0]^{.5}$$

$$= [0 + 2.5E-7 + 2.5E-5 + 0 + \ldots + 0]^{.5}$$

$$= .005 \ . \qquad (35)$$

Thus, one would expect that the error in the source's position in the x-direction could vary as much as .005.

IV

Analysis

Some of the uncertainties in the knowing the position of the event's source may be attributed in part to errors in our knowing the sensors position and the time on-board the satellite. Also, errors may be introduced into the system when there is any time delay in the reception of a signal from the event.

The process to determine the satellite's ephemeris is a multi-step process. The first step is the tracking of the satellite by four tracking stations, located in Hawaii, Alaska, and Guam and one co-located with the Master Control Station at Vandenberg AFB (24:2). These stations act basically as monitoring stations, similar to any receiver. The data is transferred to the Master Control Station (13:23).

The next step is for this data to be sent on to Naval Surface Weapon Center (NSWC) at Dahlgreen, Va. to be processed. The data consist of the accurate location of the fix sites, and the satellites ephemeris as believed by that satellite, among other information. The NSWC uses a two step process. The first is a batch process program, called CELEST, using all measurement data collected over some time span. The second method uses a recursive estimation process, via a Kalman estimator (32:78).

Then this data on the satellite's position is sent back to the Master Control Station. Each satellite's ephemeris is up loaded to the satellite along with the other normal spacecraft control commands when it comes over the Master Control Station,

approximately every 24 hours.

The satellite, then, broadcasts what it believes its own position is at that time. The monitoring stations track the satellites again and the process starts over (21:1179).

The goal of the processing is to determine the satellite's position to within 1.5 meters (one sigma) line of sight error (32:35). Currently, it has been shown that the ephemeris prediction accuracies can be expected to be within several meters (19:9).

The on-board clock will use an advanced design cesium standard clock. The expected accuracies of these clocks is to be in the range of 1 part per 10^{14} per day, resulting in an expected discrepancy of 1 second in 3,000,000 years (21:1178).

If there is a time delay in the signal reception of one millisecond, this corresponds to 300 km (1 millisecond x 3×10^{8} m/s = 3×10^{5} m = 3×10^{2} km = 300 km). This potential error is more significant than the expected error of several meters in the satellite ephemeris, or the time of the on-board clock. Thus, this investigation will concentrate on the error of propagation due to any time differences in signal reception. The primary cause for such delay, and thus uncertainty for an optical or visible sensor is cloud coverage. The amount of cloud coverage has a direct influence on how quickly the light from the event arrives at the sensor.

Two computer programs were developed to determine the expected error in uncertainty. The first program (MN) determines the location and distance (thus time) for an event using a time

difference of arrival formulation, as described in Chapter 3 (see Appendix A, page 44). The input variables are then modified on the subsequent runs. The results from the other runs are compared with the standard to find a difference, or delta. This difference is the expected error in the event location xs, ys, zs, or time of the event, ts, due to a change of the variable(s).

In the second program, Q, the standard solution from the first program is input as the position, and time (in terms of distance) of the event, along with the original positions and times for the sensors (see Appendix B, page 61). The second program is then run answering "what-if" questions. We are asking what would be the expected error if we vary some variable by some amount. As described in Chapter 3, this program utilizes the propagation of uncertainty technique, and the Jacobian matrix for solution of the partial derivatives.

The next step in the procedures is to compare the resulting differences from the two programs for a given geometry of sensors.

Case 1:

One might assume that the ideal example would be that of an "equilateral cube". Let an equilateral cube be defined as where each sensor is equidistant-distance from each other, and the distance to the source from the sensor is the same. In a coordinate system, where the center of the system is located at the source of the event, then the sensors' locations would be (10,10,20), (10,-10,20), (-10,10,20), and (-10,-10,20), where each value is multiplied by 10E3 km. When these values are plugged in for matrix $\overline{A}$ in equation (3-5), the third column is $[\ 0\ 0\ 0]^T$.

From matrix algebra, the determinant is zero, which implies that the matrix is singular. Because the matrix is singular, no inverse for matrix $\overline{A}$ exist. Thus, one is not able to solve for the source of the location, or the time of the event.

<u>Case 2</u>:

If one then varies the location of just one satellite somewhat, then a solution exist. Table 4-1 reports the results. The first column indicates the location and time for the standard sensor geometry situation where (10,10,21), (10,-10,20), (-10,10,20), and (-10,-10,20) are the locations for the respective sensors, (again, each sensor is multiplied by 10E3 km) and the distance for sensor 1 is 25,318.0 km, and the distances for the other sensors are 24,494.9 km.

Table 4-1

Case 2

	D4=0	D4=.3	D4=.03	D4=.003
MN				
XS	0	.4243E0	.3994E-1	.3703E-2
YS	0	.4243E0	.3994E-1	.3703E-2
ZS	-.1515E-2	-.1148E2	-.2539E1	-.2282E0
DS	-.1235E-2	-.3639E1	-.2113E1	-.1867E0
ΔXS		.4243E0	.3994E-1	.3703E-2
ΔYS		.4243E0	.3994E-1	.3703E-2
ΔZS		-.1148E2	-.2537E1	-.2267E0
ΔDS		-.3638E1	-.2112E1	-.1854E0
)				
ΔXS		.3674E0	.3674E-1	.3674E-2
ΔYS		.3674E0	.3674E-1	.3674E-2
ΔZS		.2241E2	.2241E1	.2241E0
ΔDS		.1830E2	.1830E1	.1830E0

[NOTE: All values are given in terms of x10E3 km.]

```
CALL VMULFM (F,FC,M,P,P,IF,IFC,RLA,IRLA,IER)
CALL MATPRT (RLA,P,P,IRLA)
PRINT*,'FC'
CALL MATPRT (FC,M,P,IFC)
RL=RLA(1,1)-1
PRINT*,'RL=    ',RL
CALL TRAMAT (RK,P,M,IRK,TRK,ITRK)
PRINT*,'TRK'
CALL MATPRT (TRK,M,P,ITRK)
CALL VMULFM (F,H,M,P,P,IF,IH,RMA,IRMA,IER)
CALL VMULFM (F,TRK,M,P,P,IF,ITRK,RMB,IRMB,IER)
PRINT*,'D =      ',D
RM=2*(RMA(1,1)-RMB(1,1)+D)
PRINT*,'RM=    ',RM
CALL VMULFM (H,HC,M,P,P,IH,IHC,RNA,IRNA,IER)
CALL MATMUL (RK,TRK,RNB,P,N,P,IRK,ITRK,IRNB)
CALL MATMUL (RK,H,RNC,P,N,P,IRK,IH,IRNC)
RN=(RNA(1,1)+RNB-(2*RNC)-(D**2))
PRINT*,'RN=   ',RN
PO=(RM**2)-(4*RL*RN)
IF (PO .LT. 0) THEN
    PRINT *,'PO IS A NEGATIVE NUMBER.'
    PO = ABS (PO)
END IF
PRINT*,'PO=   ',PO
DS1=(-RM+SQRT(PO))/(2*RL)
PRINT*,'DS1=     ',DS1
CALL SCAMAT (F,DS1,N,P,IF,FS1,IFS1)
CALL MATPRT (FS1,N,P,IFS1)
CALL MATADD (FS1,H,XS1,N,P,P,IFS1,IH,IXS1)
PRINT *,'XS1'
CALL MATPRT (XS1,N,P,IXS1)
DS2=(-RM-SQRT(PO))/(2*RL)
PRINT*,'DS2=    ',DS2
CALL SCAMAT (F,DS2,N,P,IF,FS2,IFS2)
CALL MATPRT (FS2,N,P,IFS2)
CALL MATADD (FS2,H,XS2,N,P,P,IFS2,IH,IXS2)
PRINT*,'XS2'
CALL MATPRT (XS2,N,P,IXS2)
END
```

```
A(2,2) = AFIVE
A(3,2) = ASIX
A(1,3) = ASEVEN
A(2,3) = AEIGHT
A93,3) = ANINE
READ (14,*) T1,T2,T3,T4
BONE = 2* (T2- T1)
BTWO = 2* (T3- T1)
BTHREE = 2* (T4- T1)
B(1,1) = BONE
B(2,1) = BTWO
B(3,1) = BTHREE
CT1 = (T1**2) - (T2**2)
CT2 = (T1**2) - (T3**2)
CT3 = (T1**2) - (T4**2)
CX2 = (X2**2) - (X1**2)
CX3 = (X3**2) - (X1**2)
CX4 = (X4**2) - (X1**2)
CY2 = (Y2**2) - (Y1**2)
CY3 = (Y3**2) - (Y1**2)
CY4 = (Y4**2) - (Y1**2)
CZ2 = (Z2**2) - (Z1**2)
CZ3 = (Z3**2) - (Z1**2)
CZ4 = (Z4**2) - (Z1**2)
CONE = CT1 + CX2 + CY2 + CZ2
CTWO = CT2 + CX3 + CY3 + CZ3
CTHRE = CT3 + CX4 + CY4 + CZ4
C(1,1) = CONE
C(2,1) = CTWO
C(3,1) = CTHREE
RK(1,1) = X1
RK(1,2) = Y1
RK(1,3) = Z1
D = T1
READ (14,*) N,M,P
PRINT*,'A'
CALL MATPRT (A,N,M,IA)
PRINT*,'B'
CALL MATPRT (B,N,P,IB)
PRINT*,'C'
CALL MATPRT (C,N,P,IC)
IDGT=10
CALL LINV1F (A,N,IA,AINV,IDGT,WKAREA,IER)
PRINT *, 'IER=     ',IER
PRINT *,'AINV'
CALL MATPRT (AINV,N,N,IAINV)
CALL MATMUL (AINV,B,F,M,N,P,IAINV,IB,IF)
PRINT *,'F'
CALL MATPRT (F,N,P,IF)
CALL MATMUL (AINV,C,H,M,N,P,IAINV,IC,IH)
PRINT*,'H'
CALL MATPRT (H,N,P,IH)
CALL MATCPY (F,FC,M,P,IF,IFC)
CALL MATCPY (H,HC,M,P,IH,IHC)
CALL MATCPY (RK,RKC,P,M,IRK,IRKC)
```

```
* CALLED:      PASSED:
*
* LINV1F       A,N,IA,AINV,IDGT,   FIND THE INVERSE OF A
*               WKAREA,IER          MATRIX
* MATMUL       A,B,C,N,M,P,IA,IB,  MULTIPLY TWO MATRICES
*               IC
* MATCPY       A,C,N,M,IA,IC       COPY MATRIX
* VMULFM       A,B,L,M,N,IA,IB,    MATRIX MULTIPLICATION OF
*               C,IC,IER            A TRANPOSED MATRIX BY
*                                   ANOHER MATRIX
* TRAMAT       A,N,M,IA,C,IC       TRANPOSES A MATRIX
* SCAMAT       A,Q,N,M,IA,C,IC     SCALER MULTIPLICATION
* MATADD       A,B,C,N,M,P,IA,     MATRIX ADDITION
*               IB,IC
* MATPRI       A,N,M,IA            PRINT MATRIX
*
************************************************************

      PROGRAM MN

      INTEGER N,IA,IDGT,IER,IB,IAINV,IF,IC,IH,IRK,IRKC,P,
     CIHC,IRLA,IRMA,IRMB,IRNA,IRNB,IRNC,ITRK,M,IFP1,IFP2,
     CIXS1,IXS2

      REAL A,AINV,WKAREA,B,F,C,H,RLA,RL,TRK,RMA,RMB,D,RM,
     CPO,DS1,DS2,FS2,FS1,X1,X2,FC,HC,RKC,RK,X1,X2,X3,X4,Y1,
     CRNA,RNB,RNC,RN,Y2,Y3,Y4,Z1,Z2,Z3,Z4,T1,T2,T3,T4,AONE,
     CATWO,ATHREE,AFOUR,AFIVE,ASIX,ASEVEN,AEIGHT,ANINE,
     CBONE,BTWO,BTHREE,CONE,CTWO,CTHREE,CT1,CT2,CT3,CX2,CX3,
     CCX4,CY2,CY3,CY4,CZ2,CZ3,CZ4

      DIMENSION A(4,4),B(4,4),C(4,4),AINV(4,4),WKAREA(8),
     CF(4,4),H(4,4),TRK(4,4),RK(4,4),FC(4,4),HC(4,4),
     CRMA(4,4),RMB(4,4),RNA(4,4),FS1(4,4),FS2(4,4),RKC(4,4),
     CRLA(4,4),XS1(4,4),XS2(4,4)

      PARAMETER (IA=4,IAINV=4,IB=4,IF=4,IC=4,IH=4,ITRK=4,
     CIFC=4,IHC=4,IRKC=4,IRLA=4,IRMA=4,IRMB=4,IRNA=4,IFS1=4,
     CIXS1=4,IXS2=4,IRK=4,IFS2=4)


      OPEN (14,FILE= 'DD')
      READ (14,*) X1,X2,X3,X4
      AONE = 2* (X2- X1)
      ATWO = 2* (X3- X1)
      ATHREE = 2* (X4- X1)
      READ (14,*) Y1,Y2,Y3,Y4
      AFOUR = 2* (Y2- Y1)
      AFIVE = 2* (Y3- Y1)
      ASIX =2* (Y4- Y1)
      READ (14,*) Z1,Z2,Z3,Z4
      A(1,1) = AONE
      A(2,1) = ATWO
      A(3,1) = ATHREE
      A(1,2) = AFOUR
```

```
* ATWO                REAL      2*(X3-X1)
* ATHREE              REAL      2*(X4-X1)
* AFOUR               REAL      2*(Y2-Y1)
* AFIVE               REAL      2*(Y3-Y1)
* ASIX                REAL      2*(Y4-Y1)
* ASEVEN              REAL      2*(Z2-Z1)
* AEIGHT              REAL      2*(Z3-Z1)
* ANINE               REAL      2*(Z4-Z1)
* A(4,4)              REAL      MATRIX (LEFT HAND SIDE)
* AINV(4,4)           REAL      INVERSE MATRIX
* WKAREA(4)           REAL      WORKAREA DIMENSION
* BONE                REAL      2*(T2-T1)
* BTWO                REAL      2*(T3-T1)
* BTHRE               REAL      2*(T4-T1)
* B(4,4)              REAL      MATRIX (DISTANCE DIFFERENCE)
* F(4,4)              REAL      PRODUCT MATRIX (AINV x B)
* CT1                 REAL      T1**2 - T2**2
* CT2                 REAL      T1**2 - T3**2
* CT3                 REAL      T1**2 - T4**2
* CX2                 REAL      X2**2 - X1**2
* CX3                 REAL      X3**2 - X1**2
* CX4                 REAL      X4**2 - X1**2
* CY2                 REAL      Y2**2 - Y1**2
* CY3                 REAL      Y3**2 - Y1**2
* CY4                 REAL      Y4**2 - Y1**2
* CZ2                 REAL      Z2**2 - Z1**2
* CZ3                 REAL      Z3**2 - Z1**2
* CZ4                 REAL      Z4**2 - Z1**2
* CONE                REAL      CT1 + CX2 + CY2 + CZ2
* CTWO                REAL      CT2 + CX3 + CY3 + CZ3
* CTHREE              REAL      CT3 + CX4 + CY4 + CZ4
* C(4,4)              REAL      MATRIX (SUM OF SQUARES )
* H(4,4)              REAL      PRODUCT MATRIX (AINV x C)
* RK (4,4)            REAL      MATRIX (KNOWN POSITION)
* RKC                 REAL      COPY OF RK
* FC                  REAL      COPY OF F
* HC                  REAL      COPY OF H
* RLA                 REAL      PRODUCT OF F(TRANPOSE) x F
* RL                  REAL      RLA - 1
* RMA                 REAL      PRODUCT OF F(TRANPOSE) x H
* RMB                 REAL      PRODUCT OF F(TRANPOSE) x RK
* D                   REAL      KNOWN DISTANCE
* RM                  REAL      SUM (2)
* RNA                 REAL      PRODUCT OF H(TRANPOSE) x HC
* RNB                 REAL      PRODUCT OF RK x TRK
* RNC                 REAL      PRODUCT OF RK x H
* RN                  REAL      SUM (3)
* PO                  REAL      VALUE UNDER THE SQRT SIGN
* TRK(4,4)            REAL      TRANPOSE OF RK
* DS1,DS2             REAL      VALUES OF QUADRATIC EQUATION
* FS1,FS2             REAL      SCALER PRODUCT F x DS1,DS2
* X1, X2              REAL      SUM, THE FINAL SOLUTION.
*************************************************************
*
* MODULES      ARGUEMENTS          PURPOSE:
```

```
* MULTIPLY HT BY HC USING THE IMSL LIBRARY ROUTINE VMULFM
*    (RNA)
* MULTIPLY RK BY TRK (RNB)
* MULTIPLY RK BY H (RNC)
* RN = RNA + RNB - 2 * (RNC) - (D ** 2)
* PO = (RM ** 2) - (4 * RL * RN)
* DS1 = ( - RM + SQRT(PO)) / (2 * RL)
* SCALER MULTIPLY F BY DS1 (FS1)
* ADD MATRIX FS1 TO H (XS1)
* PRINT MATRIX XS1
* DS2 = ( - RM - SQRT(PO)) / (2 * RL)
* SCALER MULTIPLY F BY DS2 (FS2)
* ADD MATRIX FS2 TO H (XS2)
* PRINT MATRIX XS2
* END
*
***********************************************************
* LOCAL VARIABLES   TYPE      PURPOSE
*
* N                 INT       ROW DIMENSION OF MATRIX
* P                 INT       COL DIMENSION OF MATRIX (B)
* IA                INT       MAX ROW DIMENSION OF A
* IDGT              INT       INPUT OPTION (LINV1F)
* IER               INT       ERROR STATEMENT
* IB                INT       MAX ROW DIMENSION OF B
* IAINV             INT       MAX ROW DIMENSION OF AINV
* IF                INT       MAX ROW DIMENSION OF F
* IC                INT       MAX ROW DIMENSION OF C
* IH                INT       MAX ROW DIMENSION OF H
* IRK               INT       MAX ROW DIMENSION OF RK
* IFC               INT       MAX ROW DIMENSION OF FC
* IHC               INT       MAX ROW DIMENSION OF HC
* IRKC              INT       MAX ROW DIMENSION OF RKC
* IRLA              INT       MAX ROW DIMENSION OF RLA
* IRMA              INT       MAX ROW DIMENSION OF RMA
* IRMB              INT       MAX ROW DIMENSION OF RMB
* IRNA              INT       MAX ROW DIMENSION OF RNA
* IRNB              INT       MAX ROW DIMENSION OF RNB
* IRNC              INT       MAX ROW DIMENSION OF RNC
* ITRK              INT       MAX ROW DIMENSION OF TRK
* IFP1              INT       MAX ROW DIMENSION OF FP1
* IFP2              INT       MAX ROW DIMENSION OF FP2
* IXS1              INT       MAX ROW DIMENSION OF XS1
* IXS2              INT       MAX ROW DIMENSION OF XS2
* M                 INT       COL DIMENSION OF MATRIX (A)
*
* X1-4              REAL      LOCATION IN THE X-DIRECTION
*                              FOR THE APPROPRIATE SENSOR
* Y1-4              REAL      LOCATION IN THE Y-DIRECTION
*                              FOR THE APPROPRIATE SENSOR
* Z1-4              REAL      LOCATION IN THE Z-DIRECTION
*                              FOR THE APPROPRIATE SENSOR
* T1-4              REAL      DISTANCE AWAY FOR THE
*                              APPROPRIATE SENSOR
* AONE              REAL      2*(X2-X1)
```

Appendix A

Program MN

```
***********************************************************
*
* MAIN MODULE: MN
*
* PROJECT: THESIS                    DATE: 1 OCT 84
* PROGRAMMER: C. M. WOZNIAKOWSKI
**********************************************************
*
* MODULE DESCRIPTION: GIVEN THE POSITION OF FOUR SENSORS
*    AND THE DISTANCE FROM AN EVENT, CALCULATE THE ELEMENTS
*    OF THE MATRICES A, B, AND C, THROUGH THE USE OF TIME
*    DIFFERENCE OF ARRIVAL EQUATIONS, WHICH FORM A LINEAR
*    MATRIX EQUATION (AX = BD + C).  THEN SOLVE FOR X AND D
*    (THE LOCATION AND DISTANCE (TIME) OF THE EVENT), USING
*    THE IMSL SUBROUTINES LINV1F AND VMULFM.
*
**********************      NOTE      *********************
*
*    THE INPUT FILE, DD, CONSISTS OF X1, X2, X3, AND X4 IN
* THAT ORDER  IN THE FIRST ROW.  THE SECOND ROW IS MADE UP
* OF Y'S, WHILE THE THIRD ROW IS MADE UP OF Z'S, AND THE
* FOUTH OF DISTANCES.  THE FINAL ROW CONSISTS OF THE ROW
* AND COLUMN DIMENSIONS FOR THE RESPECTED MATRICES.
*    INSURE THAT IMSL IS ATTACHED AND THAT ALL FILES ARE
*  REWOUND BEFORE EACH RUN.
***********************************************************
* DESCRIPTION OF ALGORITHM DEVELOPMENT:
*
* READ IN THE XI,YI,AND ZI VALUES FOR THE SENSOR(I)
* CALCULATE THE ELEMENTS OF MATRIX A
* READ IN THE TI (DISTANCE) VALUES FOR THE SENSOR(I)
* COMPUTE THE ELEMENTS OF MATRICES B AND C
* READ IN N, M, AND P
* RK (x,y,z OF A SENSOR) = [X1, Y1, Z1]
* D = T1
* COMPUTE THE INVERSE OF A USING THE IMSL LIBRARY ROUTINE
*    LINV1F (AINV)
* PRINT IER
* MULTIPLY AINV BY B TO GET MATRIX F
* MULTIPLY AINV BY C TO GET MATRIX H
* COPY MATRIX F (FC), H (HC), AND RK (RKC)
* MULTIPLY FT BY F USING THE IMSL LIBRARY ROUTINE VMULFM
*    (RLA)
* RL = RLA - 1
* TRANPOSE THE MATRIX RK (TRK)
* MULTIPLY FT BY H USING THE IMSL LIBRARY ROUTINE VMULFM
*    (RMA)
* MULTIPLY FT BY RK USING THE IMSL LIBRARY ROUTINE VMULFM
*    (RMB)
* RM = 2 * (RMA -RMB + D)
```

the program is required to verify the programs and confirm the results herein.

V

Conclusions

In summary, the results indicated that the geometry for viewing an event is a critical factor. In the case of the "equilateral cube", the procedure used here cannot produce a solution, because $\bar{A}$ is a singular matrix and $\bar{A}^{-1}$ does not exist. When the sensors are more dispersed and random in location, the procedure generally gives accurate location results.

Another result from these example calculations is that the determined locations are more sensitive to in the zs (altitude) than in xs or ys because of uncertainties in the distance (time of arrival) to one or more of the sensors In fact, the expected error in the zs approaches a factor of 10 times that in xz and ys This may be partially explained by the fact that the sensors are all on the same side of the event, bias the results. Future research might study more well behaved formulas that takes this fact into consideration. Hence, future studies might look at the situation where ground- and spaced-based sensors are working in conjunction with each other to determine the height.

Tables 4-1 and 4-2 indicate that the expected errors are monotonic. By monotonic, I mean if the error for only one sensor is reduced, then the resulting expected error is also reduced, and the accuracy of location is improved.

Future areas of inquiry might include working the same the same problem but in a different coordinate system, spherical or geocentric coordinate system for example. Also, more exercise of

values from "Q", though they did not match.

Case 6:

The next example employing the same sensor configuration as in Case 3 that one might compare is when one changes the same sensor by a +.3E3 and a -.3E3 km, and then ±.03E3 km. The resulting values for the second program remains the same when the input variable is a "+" or a "-" because the individual inputs are being squared. In the first program, the expected error for +.3E3 km and -.3E3 km runs agree with each other to one significant number. Likewise, when the distance is varied by a +.03E3 km and -.03E3 km, the differences agree to one significant number. And both cases agree with the expected differences from "Q" in a similar manner.

Case 5:

When one modified the distances by a ± .3E3 km in some combination, for the same geometry as in Case 3, one came closer to approximate the values garned from program "Q". In example 1, all the distances are varied by a +.3E3 km. In example 2, the distances for sensors one and two are varied by +.3E3 km, while the distances for sensors are varied by a -.3E3 km. In example 3, the distance for sensor one is perturbed by a +.3E3 km, while the other senors´ distances are perturbed by a -.3E3 km. In example 3, we came close to a singularity. And as before, we had to force the solution. The results more closely approximated the results in "Q". But again, the results would not satisfy the set of equations (3-1) and could not be validated.

Though not represented in any table, the same is true in the case of two sensors. When one varied two sensors by a +.3E3 and -.3E3 km, the resulting delta´s more closely approaches those

Table 4-5

Case 6

	D4=0	D4=.3	D4=-.3	D4=.03	D4=-.03
MN					
X	-.1172E-5	-.7492E-1	.6801E-1	-.6882E-2	.6826E-2
Y	-.2348E-5	-.5009E0	.5681E0	-.5155E-1	.5224E-1
Z	.2140E-4	.6396E1	-.6921E1	.6492E0	-.6478E0
D	.1608E-4	.4508E1	-.5623E1	.4889E0	-.5000E0
ΔXS		-.7492E-1	.6802E-1	-.6881E-2	.6827E-2
ΔYS		-.5009E0	.5681E0	-.5154E-1	.5224E-1
ΔZS		.6396E1	-.6921E1	.6420E0	-.6478E0
ΔDS		.4508E1	-.5623E1	.4889E0	-.5000E0
Q					
ΔXS		.7773E-1		.7773E-2	
ΔYS		.5432E0		.5432E-1	
ΔZS		.6641E1		.6641E0	
ΔDS		.5091E1		.5091E0	

perturbed then when one sensor is perturbed, unless the perturbations are in such a direction that they cancel each other out. Such wasthe case for four sensors. The only difference is that the distance is .3E3 km larger for each sensor. Thus, one gets the same values for xs, ys, and zs and a delta of zero for each. And one gets a delta of .3E3 km for the distance error, the same input difference that each was varied by.

Another indication from this example is that the determined locations are more sensitive in the zs than in the xs or ys. In fact the expected error in the zs approaches a factor of ten. This may be partially explained by the fact that the sensors are all on the same side of the event, and bias the results.

Table 4-4

Case 5

	D4=0	EX.1	EX.2	EX.3
MN				
X	-.1172E-5	-.1172E-5	-.6565E0	-.5451E0
Y	-.2348E-5	-.2348E-5	-.3382E0	-.1364E1
Z	.2140E-4	.2140E-4	.3253E1	.8451E1
D	.1608E-4	.1608E-4	.2347E1	.8307E1
ΔXS		0	-.6565E0	-.5451E0
ΔYS		0	-.3382E0	-.1364E1
ΔZS		0	.3253E1	.8451E1
ΔDS		.3000E0	.2347E1	.8307E1
Q				
ΔXS		.7314E0		
ΔYS		.2042E1		
ΔZS		.1305E2		
ΔDS		.1365E2		

km. Then we varied two sensors by the same factor, then three and finally four sensors. The results are in Table 4-3. The same geometry as in Case 3 is used once more.

Again the values for the differences from the second program are larger than the first program. Thus the results reiterate the idea that the program "Q" forms an envelope in which the results from the program "MN" reside in.

One should be able to discern that when an even number of sensors are altered in "MN" by the same amount, then the values for the differences are significantly less then those differences from "Q". This can be explained as some form of symmetry is produced, which in turn reduces the differences.

When an odd number of sensors are varied in "MN" by the same amount, then no symmetry exists. Also, the error of uncertainty is larger when two, three, or four sensors are

Table 4-3

Case 4

	D4=0	1SENSOR	2SENSORS	3SENSORS	4SENSORS
MN					
X	-.1172E-5	-.7492E-1	.3864E-1	.5991E0	-.1172E-5
Y	-.2348E-5	-.5009E0	.5555E0	.1552E1	-.2348E-5
Z	.2140E-4	.6396E1	-.1668E1	-.1147E2	.2140E-4
D	.1608E-4	.4508E1	-.1143E1	-.9265E1	.3000E0
ΔXS		-.7492E-1	.3864E0	.5991E0	0
ΔYS		-.5009E0	.5555E0	.1552E1	0
ΔZS		.6396E1	-.1668E1	-.1147E2	0
ΔDS		.4508E1	-.1143E1	-.9265E1	.3000E0
Q					
ΔXS		.7773E-1	.1436E0	.5059E0	.7314E0
ΔYS		.5432E0	.1217E1	.1487E1	.2042E1
ΔZS		.6641E1	.1070E2	.1422E2	.1805E2
ΔDS		.5091E1	.8074E1	.1069E2	.1365E2

The values in Table 4-1 in columns three and four are determined by only varying the distance in the fourth sensor by .03 and .003, respectively. The expected error of propagation tends to agree in both programs in magnitude to within 10 percent. The second method is approximate and always yield a positive value. Also, the data indicates the expected errors are monotonic when the input variables are altered in a similar manner.

Case 3:

In this case, the position of the sensors are varied and do not approach a "equilateral cube". The position for the sensors are (12,10,21), (-11,11,20.5), (15,-10,20) and (-16,-15,20) in a similarly defined coordinate system. The distances for the respective sensors are 26,172.5 km, 25,734.2 km, 26,925.8 km and 29,681.6 km. Again, the distances for the fourth sensor are pertubed by .3, .03, and .003 E3 km, while everything remains constant. In another words, the times are being perturbed by 1, .1, and .01 millisecond.

The values for the respective delta´s decrease by a factor of approximately 10, each time, in each program. Also, these results agree with the previous examples in that the expected error is monotonic.

As it happened previously, the second program has larger values for the resultant delta´s. The delta values for each program do agree to the first significant digit.

Case 4:

Next, we varied the distance in one sensor by a factor of .3E3

The second column represents the instance when the distance D4 for one of the three sensors at 24,494.9 km is changed to 24,794.9 km. This represents a shift D4 of 300 km, corresponding to a time shift of 0.001 sec. Initial calculations give non-real results in the first program, when one is trying to solve for the time of the event, DS1 and DS2, one is taking a square root of a negative number. However, when one forces a solution by taking the square root of the absolute value of the number, in column two, one then obtains the results as indicated in Table 4-1. The solution is unable to be validated in the set of equations (3-1). This indicates that the geometry of the sensors is as important when looking down as it is when trying to determine one's own position (6:37, 30:10-14).

Table 4-2

Case 3

	D4=0	D4=.3	D4=.03	D4=.003
MN				
XS	-.1172E-5	-.7492E-1	-.6882E-2	-.6865E-3
YS	-.2348E-5	-.5009E0	-.5155E-1	-.5187E-2
ZS	.2140E-4	.6396E1	.6420E0	.6446E-1
DS	.1608E-4	.4508E1	.4889E0	.4939E-1
ΔXS		-.7492E-1	-.6881E-2	-.6854E-3
ΔYS		-.5008E-1	-.5154E-1	-.5185E-2
ΔZS		.6396E1	.6420E0	.6444E-1
ΔDS		.4508E1	.4889E0	.4938E-1
Q				
ΔXS		.7773E-1	.7773E-2	.7773E-3
ΔYS		.5432E0	.5432E-1	.5432E-2
ΔZS		.6641E1	.6641E0	.6641E-1
ΔDS		.5091E1	.5901E0	.5901E-1

```
************************************************************
*
* SUBROUTINE NAME:  MATMUL
*
* ARGUEMENT LIST: A,B,C,N,M,P,IA,IB,IC
* CALLED BY:   MM
* PROJECT: THESIS              DATE: 1 OCT 84
* PROGRAMMER: C. M. WOZNIAKOWSKI
*
************************************************************
*
* MODULE DESCRIPTION: MULTIPLY TWO MATRICES
*
************************************************************
*
* ARGUEMENTS  IN/  TYPE  PASSED/ PURPOSE
* NAME:       OUT        GLOBAL
*
* A,B         IN   REAL  PASSED  MATRICES TO BE MULTIPLIED
* C           OUT  REAL  PASSED  THE PRODUCT MATRIX
* N           IN   INT   PASSED  ROW DIMENSION OF A
* M           IN   INT   PASSED  ROW DIMENSION OF B
* P           IN   INT   PASSED  COLUMN DIMENSION OF B
* IA,IB,IC    IN   INT   PASSED  MAX ROW DIMENSIONS
*
************************************************************
*
* DESCRIPTION OF ALGORITHM DEVELOPMENT:
*
* FOR  (I=1,N) LOOP
*    FOR (J=1,P) LOOP
*         SUM = 0
*         FOR (K=1,M) LOOP
*              SUM=SUM+(A(I,K)*(B(K,J)))
*         END LOOP
*         C(I,J)=SUM
*    END LOOP
* END LOOP
* END
*
************************************************************
*
* LOCAL VARIABLES   TYPE   PURPOSE
*
* I,J,K             INT    COUNTING VARIABLES
* SUM               REAL   SUM OF ROW x COL MULTIPLICATION
************************************************************

      SUBROUTINE MATMUL (A,B,C,N,M,P,IA,IB,IC)

      INTEGER N,M,P,IA,IB,IC,I,J,K
      REAL A,B,C,SUM
      DIMENSION A(4,4),B(4,4),C(4,4)

      DO 71 I=1,N
```

```
          DO 81 J=1,P
               SUM=0
               DO 91 K=1,M
                    SUM=SUM+(A(I,K)*B(K,J))
91             CONTINUE
               C(I,J)=SUM
81        CONTINUE
71    CONTINUE
      END
```

```
************************************************************
*
* SUBROUTINE NAME: MATPRT
*
* ARGUEMENT LIST: A,N,M,IA
* CALLED BY: MM
* PROJECT: THESIS                    DATE: 1 OCT 84
* PROGRAMMER: C. M. WOZNIAKOWSKI
*
************************************************************
*
* MODULE DESCRIPTION: PRINT A MATRIX BY ROWS
*
************************************************************
*
* AGRUEMENTS IN/  TYPE  PASSED/  PURPOSE
* NAME       OUT        GLOBAL
*
* A(N,M)     I/O  REAL  PASSED   THE MATRIX ELEMENTS
* N          IN   INT   PASSED   ROWS IN MATRIX
* M          IN   INT   PASSED   COLS IN MATRIX
* IA         IN   INT   PASSED   MAX ROW DIMENSION
* STD OUTPUT OUT  TEXT  GLOBAL   OUTPUT MATRIX
*
************************************************************
*
* DESCRIPTION OF ALGORITHM DEVELOPMENT:
*
* PRINT BLANK LINE
* FOR I=1,N LOOP
*    FOR J=1,M LOOP
*         PRINT (FORMAT) MATRIX A(I,J)
*    END LOOP
* END LOOP
* END
*
************************************************************
*
* LOCAL VARIABLES  TYPE  PURPOSE
*
**************************************************************************

      SUBROUTINE MATPRT (A,N,M,IA)

      INTEGER N,M,IA,I,J
      REAL A(IA,M)

      PRINT *
      DO 31 I=1,N
          PRINT *
          PRINT 40,(A(I,J),J=1,M)
 40       FORMAT (5E17.9)
 31   CONTINUE
      END
```

```
***********************************************************
*
* SUBROUTINE NAME : MATCPY
*
* ARGUEMENT LIST: A,C,N,M,IA,IC
* CALLED BY: MM
* PROJECT: THESIS                    DATE: 1 OCT 84
* PROGRAMMER: C. M. WOZNIAKOWSKI
*
***********************************************************
*
* MODULE DESCRIPTION: COPIES (STORES) A MATRIX IN ANOTHER
*    LOCATION
*
***********************************************************
*
* AGRUEMENTS    IN/  TYPE  PASSED/  PURPOSE
* NAME          OUT        GLOBAL
*
* A(IA,M)       IN   REAL  PASSED   MATRIX TO BE COPIED
* C(IC,M)       OUT  REAL  PASSED   COPIED MATRIX
* N             IN   INT   PASSED   ROW DIMENSION
* M             IN   INT   PASSED   COL DIMENSION
* IA            IN   INT   PASSED   MAX ROW DIMENSION FOR A
* IC            IN   INT   PASSED   MAX ROW DIMENSION FOR C
*
***********************************************************
*
* DESCRIPTION OF ALGORITHM DEVELOPMENT:
*
* FOR (I=1,N) LOOP
*    FOR (J=1,M) LOOP
*         C(I,J) = A(I,J)
*    END LOOP
* END LOOP
* END
*
***********************************************************
*
* LOCAL VARIABLES  TYPE  PURPOSE
*
* I,J              INT   COUNTING VARIABLES
*
***********************************************************

      SUBROUTINE MATCPY (A,C,N,M,IA,IC)

      INTEGER N,M,IA,IC,I,J
      REAL A,C
      DIMENSION A(4,4),C(4,4)

      DO 51 I=1,N
          DO 61 J=1,M
               C(I,J)=A(I,J)
 61       CONTINUE
```

```
51   CONTINUE
     END
```

```
***********************************************************
*
* SUBROUTINE NAME: TRAMAT
*
* ARGUEMENT LIST: A,N,M,IA,IC,C
* CALLED BY: MM
* PROJECT: THESIS                     DATE: 1 OCT 84
* PROGRAMMER: C. M. WOZNIAKOWSKI
*
***********************************************************
*
* MODULE DESCRIPTION: TRANPOSES A MATRIX
*
***********************************************************
*
* AGRUEMENT    IN/  TYPE  PASSED/  PURPOSE
* NAME         OUT        GLOBAL
*
* A(IA,N)      IN   REAL  PASSED   MATRIX TO BE COPIED
* C(IC,M)      OUT  REAL  PASSED   THE TRANPOSED MATRIX
* N            IN   INT   PASSED   ROW DIMENSION
* M            IN   INT   PASSED   COL DIMENSION
* IA           IN   INT   PASSED   MAX ROW DIMENSION OF A
* IC           IN   INT   PASSED   MAX ROW DIMENSION OF C
*
***********************************************************
*
* DESCRIPTION OF ALGORITHM DEVELOPMENT:
*
* FOR (I=1,N) LOOP
*    FOR (J= 1,M) LOOP
*         C(J,I)=A(I,J)
*    END LOOP
* END LOOP
* END
*
***********************************************************
*
* LOCAL VARIABLES   TYPE  PURPOSE
*
*  I,J              INT   COUNTING VARIABLES
*
***********************************************************

      SUBROUTINE TRAMAT (A,N,M,IA,C,IC)

      INTEGER N,M,IA,IC,I,J
      REAL A,C
      DIMENSION A(4,4),C(4,4)

      DO 65 I=1,N
          DO 64 J=1,M
               C(J,I)=A(I,J)
 64       CONTINUE
```

```
65    CONTINUE
      END
```

```
************************************************************
*
* SUBROUTINE NAME: SCAMAT
*
* ARGUEMENT LIST: A,Q,N,M,IA,C,IC
* CALLED BY: MN
* PROJECT: THESIS                    DATE: 1 OCT 84
* PROGRAMMER: C. M. WOZNIAKOWSKI
*
************************************************************
*
* MODULE DESCRIPTION: MULTIPLY A MATRIX BY A SCALER
*
************************************************************
*
* ARGUEMENT    IN/  TYPE PASSED/   PURPOSE
* NAME         OUT       GLOBAL
*
* A(IA,N)      IN   REAL PASSED    MATRIX TO BE MULTIPLIED
* Q            IN   REAL PASSED    SCALER
* C(IC,M)      OUT  REAL PASSED    THE PRODUCT MATRIX
* N            IN   INT  PASSED    ROW DIMENSION
* M            IN   INT  PASSED    COL DIMENSION
* IA           IN   INT  PASSED    MAX ROW DIMENSION FOR A
* IC           OUT  INT  PASSED    MAX ROW DIMENSION FOR C
*
************************************************************
*
* DESCRIPTION OF ALGORITHM DEVELOPMENT:
*
* FOR (I=1,N) LOOP
*    FOR (J=1,M) LOOP
*         C(I,J) = Q * A(I,J)
*    CONTINUE
* CONTINUE
* END
*
************************************************************
*
* LOCAL VARIABLES   TYPE      PURPOSE
*
*  I,J              INT       COUNTING VARIABLES
*
************************************************************

      SUBROUTINE SCAMAT (A,Q,N,M,IA,C,IC)

      INTEGER IA,IC,N,M,I,J
      REAL Q,A,C
      DIMENSION A(4,4),C(4,4)

      DO 54 I=1,N
          DO 53 J=1,M
               C(I,J) = Q*(A(I,J))
 53       CONTINUE
```

```
54    CONTINUE
      END
```

```
************************************************************
*
* SUBROUTINE NAME: MATADD
*
* ARGUEMENT LIST: A,B,C,N,M,P,IA,IB,IC
* CALLED BY: MN
* PROJECT: THESIS                    DATE: 1 OCT 84
* PROGRAMMER: C. M. WOZNIAKOWSKI
*
************************************************************
*
* MODULE DESCRIPTION: ADD TWO MATRICES
*
************************************************************
*
* ARGUEMENT    IN/  TYPE PASSED/   PUPOSE
* NAME         OUT       GLOBAL
*
* A,B          IN   REAL PASSED    MATRICES TO BE ADDED
* C            OUT  REAL PASSED    THE SUM
* N            IN   INT  PASSED    ROW DIMENSION OF A
* M            IN   INT  PASSED    ROW DIMENSION OF B
* P            IN   INT  PASSED    COL DIMENSION OF B
* IA           IN   INT  PASSED    MAX ROW DIMENSION OF A
* IB           IN   INT  PASSED    MAX ROW DIMENSION OF B
* IC           IN   INT  PASSED    MAX ROW DIMENSION OF C
*
************************************************************
*
* DESCRIPTION OF ALGORITHM DEVELOPMENT:
*
* FOR (I=1,N) LOOP
*    FOR (J=1,M) LOOP
*         C(I,J) = A(I,J) + B(I,J)
*    END LOOP
* END LOOP
* END
*
************************************************************
*
* LOCAL VARIABLES   TYPE      PURPOSE
*
*    I,J            INT       COUNTING VARIABLES
*
************************************************************

      SUBROUTINE MATADD (A,B,C,N,M,P,IA,IB,IC)

      INTEGER N,M,P,IA,IB,IC,I,J
      REAL A,B,C
      DIMENSION A(4,4),B(4,4),C(4,4)

      DO 23 I=1,N
          DO 24 J=1,M
               C(I,J)=A(I,J)+B(I,J)
```

```
24        CONTINUE
23    CONTINUE
      END
```

Appendix B

Program Q

```
************************************************************
*
* MAIN MODULE: Q
*
* PROJECT:     THESIS              DATE: 15 OCT 84
* PROGRAMMER: C. M. WOZNIAKOWSKI
*
************************************************************
*
* MODULE DESCRIPTION: GIVEN THE POSITION AND DISTANCE (TIME)
*    FOR FOUR SENSORS AND THE SOURCE FOR SOME EVENT,
*    DETERMINE THE EXPECTED ERROR OF PROPGATION IN THE X-,
*    Y-, AND Z-DIRECTION AND TIME BY MEANS OF IMPLICIT
*    DIFFERENTIATION AND JACOBIAN MATRICES.  THIS PROGRAM
*    USES THE IMSL SUBROUTINE 'LINV3F'.
*
************************************************************
*
* DESCRIPTION OF ALGORITHM DEVELOPMENT:
*
* READ IN THE X VALUES FOR THE SOURCE AND THE SENSORS
* READ IN THE Y VALUES FOR THE SOURCE AND THE SENSORS
* READ IN THE Z VALUES FOR THE SOURCE AND THE SENSORS
* READ IN THE T VALUES FOR THE SOURCE AND THE SENSORS
* CALCULATE THE ELEMENTS OF MATRIX D
* FIND THE DETERMINANT OF D USING THE IMSL LIBRARY ROUTINE
*    LINV3F (DD)
* COPY THE MATRIX D TWICE (AA, DF)
* FIND THE PARTIAL DERIVATIVE OF XS WITH RESPECT TO X1->X4
*    THROUGH THE USE OF JACOBIAN MATRIX (DET)
* SQUARE THE PARTIAL DERIVATIVES
* MULTIPLY THE SQUARE PARTIAL DERIVATIVES BY A SQUARE OF THE
*    DELTA FOR THE RESPECTIVE XI
* REPEAT THE PROCESS FOR YI, ZI, AND TI
* SUM THE PRODUCTS
* TAKE THE SQUARE ROOT OF THE SUM TO FIND A DELTA XS
* REPAET THE PROCESS TO FIND DELTA YS, ZS, AND TS
*
*************************  NOTE   ************************
*
*    THE INPUT FILE, DT1, CONSIST OF XS,X1,X2,X3, AND X4 IN
* THAT ORDER FOR THE FIRST ROW.  THE NEXT ROW CONSIST OF
* THE YI'S; THEN THE ZI'S MAKE UP THE FOLLOWING ROW, AND
* FINALLY THE TI'S.  THE FINAL ROW IN THE FILE CONSIST OF
* THE ROW AND COLUMN DIMENSIONS FOR THE MATRICES INVOLVED.
* THE INPUT FILE OF DEL IS MADE UP OF A SINGLE COLUMN MATRIX
* LISTING THE RESPECTED DEL'S AS THEY ARE TO BE USED.
*    ALSO INSURE THAT ALL FILES ARE REWOUND, AND THAT IMSL
* IS ATTACHED BEFORE RUNNING THE PROGRAM.
*
```

```
*********************************************************
*
* LOCAL        TYPE:     PURPOSE:
* VARIABLES:
*
* IJOB         INT       INPUT OPTION PARAMETER
* IER          INT       ERROR OPTION
* N            INT       ROW DIMENSION OF MATRIX
* ID           INT       MAX ROW DIMENSION FOR MATRIX D
* IAA          INT       MAX ROW DIMENSION FOR MATRIX AA
* IBB          INT       MAX ROW DIMENSION FOR MATRIX BB
* ICC          INT       MAX ROW DIMENAION FOR MATRIX CC
* ITT          INT       MAX ROW DIMENSION FOR MATRIX TT
* M            INT       COL DIMENSION FOR MATRIX
* LV,I         INT       COUNTING VARIABLES
* IDF          INT       MAX ROW DIMENSION FOR MATRIX DF
*
* AONE         REAL      ELEMENT (1,1) OF MATRIX D
* ATWO         REAL      ELEMENT (2,1) OF MATRIX D
* ATHREE       REAL      ELEMENT (3,1) OF MATRIX D
* AFOUR        REAL      ELEMENT (4,1) OF MATRIX D
* BONE         REAL      ELEMENT (1,2) OF MATRIX D
* BTWO         REAL      ELEMENT (2,2) OF MATRIX D
* BTHREE       REAL      ELEMENT (3,2) OF MATRIX D
* BFOUR        REAL      ELEMENT (4,2) OF MATRIX D
* CONE         REAL      ELEMENT (1,3) OF MATRIX D
* CTWO         REAL      ELEMENT (2,3) OF MATRIX D
* CTHREE       REAL      ELEMENT (3,3) OF MATRIX D
* CFOUR        REAL      ELEMENT (4,3) OF MATRIX D
* EONE         REAL      ELEMENT (1,4) OF MATRIX D
* ETWO         REAL      ELEMENT (2,4) OF MATRIX D
* ETHREE       REAL      ELEMENT (3,4) OF MATRIX D
* EFOUR        REAL      ELEMENT (4,4) OF MATRIX D
* X1-X4        REAL      X-POSITION FOR SENSOR I
* Y1-Y4        REAL      Y-POSITION FOR SENSOR I
* Z1-Z4        REAL      Z-POSITION FOR SENSOR I
* T1-T4        REAL      DISTANCES FOR SENSOR I
* XS,YS,ZS     REAL      LOCATION OF SOURCE
* TS           REAL      DISTANCE(TIME) OF SOURCE
* DELX1-4      REAL      DIFFERENCES IN X-DIRECTION FOR
*                         SENSOR I
* DELY1-4      REAL      DIFFERENCES IN Y-DIRECTION FOR
*                         SENSOR I
* DELZ1-4      REAL      DIFFERENCES IN Z-DIRECTION FOR
*                         SENSOR I
* DELT1-4      REAL      DIFFERENCES IN DISTANCES FOR SENSOR I
* DELXS        REAL      ESTIMATED ERROR IN X-DIRECTION
* DELYS        REAL      ESTIMATED ERROR IN Y-DIRECTION
* DELZS        REAL      ESTIMATED ERROR IN Z-DIRECTION
* DELTS        REAL      ESTIMATED ERROR IN DISTANCE
* DN           REAL      DETERMINANT
* SUMXX        REAL      SUM OF PRODUCTS INVOLVING DELXI'S
*                         AND PART. DER. OF XS
* SUMXY        REAL      SUM OF PRODUCTS INVOLVING DELYI'S
*                         AND PART. DER. OF XS
```

```
* SUMXZ          REAL      SUM OF PRODUCTS INVOLVING DELZI'S
*                           AND PART. DER. OF XS
* SUMXT          REAL      SUM OF PRODUCTS INVOLVING DELTI'S
*                           AND PART. DER. OF XS
* SUMX           REAL      SUM OF ALL PRODUCTS INVOLVING PART.
*                           DER. OF XS
* A(16)          REAL      PARTIAL DERIVATIVES WRT XI
* B(16)          REAL      PARTIAL DERIVATIVES WRT YI
* C(16)          REAL      PARTIAL DERIVATIVES WRT ZI
* D(16)          REAL      PARTIAL DERIVATIVES WRT TI
* WKAREA         REAL      WORKAREA DIMENSION
* DF             REAL      MATRIX COPY OF D
* AA             REAL      JACOBIAN MATRIX FOR XS
* BB             REAL      JACOBIAN MATRIX FOR YS
* CC             REAL      JACOBIAN MATRIX FOR ZS
* TT             REAL      JACOBIAN MATRIX FOR TS
* PXSX1-X4       REAL      PART. DER. OF XS WRT X1-X4
* PXSY1-Y4       REAL      PART. DER. OF XS WRT Y1-Y4
* PXSZ1-Z4       REAL      PART. DER. OF XS WRT Z1-Z4
* PXST1-T4       REAL      PART. DER. OF XS WRT T1-T4
* PXX1-X4        REAL      PRODUCT OF THE SQUARES  OF  PART.
*                           DER. AND THE DELX1-X4
* PXY1-Y4        REAL      PRODUCT OF THE SQUARES  OF  PART.
*                           DER. AND THE DELY1-Y4
* PXZ1-Z4        REAL      PRODUCT OF THE SQUARES OF PART.
*                           DER. AND THE DELZ1-Z4
* PXT1-T4        REAL      PRODUCT OF THE SQUARES OF PART.
*                           DER. AND THE DELT1-T4
* PYSX1-X4       REAL      PART. DER  OF YS WRT X1-X4
* PYSY1-Y4       REAL      PART. DER. OF YS WRT Y1-Y4
* PYSZ1-Z4       REAL      PART. DER. OF YS WRT Z1-Z4
* PYST1-T4       REAL      PART. DER. OF YS WRT T1-T4
* PYX1-X4        REAL      PRODUCT OF THE SQUARES OF PART.
*                           DER. AND THE DELX1-X4
* PYY1-Y4        REAL      PRODUCT OF THE SQUARES OF PART.
*                           DER. AND THE DELY1-Y4
* PYZ1-Z4        REAL      PRODUCT OF THE SQUARES OF PART.
*                           DER. AND THE DELZ1-Z4
* PYT1-T4        REAL      PRODUCT OF THE SQUARES OF PART.
*                           DER. AND THE DELT1-T4
* SUMYX          REAL      SUM OF PRODUCTS INVOLVING DELXI'S
*                           AND PART. DER. OF YS
* SUMYY          REAL      SUM OF PRODUCTS INVOLVING DELYI'S
*                           AND PART. DER. OF YS
* SUMYZ          REAL      SUM OF PRODUCTS INVOLVING DELZI'S
*                           AND PART. DER. OF YS
* SUMYT          REAL      SUM OF PRODUCTS INVOLVING DELTI'S
*                           AND PART. DER. OF YS
* SUMY           REAL      SUM OF ALL PRODUCTS INVOLVING PART.
*                           DER. OF YS
* PZSX1-X4       REAL      PART. DER. OF ZS WRT X1-X4
* PZSY1-Y4       REAL      PART. DER. OF ZS WRT Y1-Y4
* PZSZ1-Z4       REAL      PART. DER. OF ZS WRT Z1-Z4
* PZST1-T4       REAL      PART. DER. OF ZS WRT T1-T4
* PZX1-X4        REAL      PRODUCT OF THE SQUARES OF PART.
```

```
        LV=LV+1
        CC(I,3)=E(LV)
460   CONTINUE
      CALL DET (CC,DF,DN,IDF,ICC)
      PZST2=DN/DD
      READ (14,*) DELT2
      PZT2=(PZST2**2)*(DELT2**2)
      PRINT*,'PZT2=  ',PZT2
      DO 470 I=1,4
        LV=LV+1
        CC(I,3)=E(LV)
470   CONTINUE
      CALL DET (CC,DF,DN,IDF,ICC)
      PZST3= DN/DD
      READ (14,*) DELT3
      PZT3=(PZST3**2)*(DELT3**2)
      PRINT*,'PZT3=    ',PZT3
      DO 480 I=1,4
        LV=LV+1
        CC(I,3)=E(LV)
480   CONTINUE
      CALL DET (CC,DF,DN,IDF,ICC)
      PZST4=DN/DD
      READ (14,*) DELT4
      PZT4=(PZST4**2)*(DELT4**2)
      PRINT*,'PZT4=    ',PZT4
      SUMZT=PZT1+PZT2+PZT3+PZT4
      PRINT*,'SUMZT =        ',SUMZT
      SUMZ=SUMZX+SUMZY+SUMZZ+SUMZT
      PRINT*,' SUMZ =       ',SUMZ
      DELZS=SQRT(SUMZ)
      PRINT*,'DELZS =        ',DELZS
      CALL MATCPY (DF,TT,N,M,IDF,ITT)
      LV =0
      DO 490 I=1,4
        LV = LV + 1
        TT(I,4) = A(LV)
490   CONTINUE
      CALL DET(TT,DF,DN,IDF,ITT)
      PTSX1 =  DN/DD
      PRINT*,'PTSX1=     ',PTSX1
      READ (14,*)  DELX1
      PTX1 = (PTSX1 ** 2) * (DELX1 ** 2)
      PRINT*,'PTX1 =     ',PTX1
      LV=4
      DO 500 I=1,4
        LV = LV + 1
        TT(I,4) = A(LV)
500   CONTINUE
      CALL DET (TT,DF,DN,IDF,ITT)
      PTSX2 = DN/DD
      READ (14,*) DELX2
      PTX2 = (PTSX2**2)*(DELX2**2)
      PRINT*,'PTX2=      ',PTX2
      LV=8
```

```
      SUMZY=PZY1+PZY2+PZY3+PZY4
      PRINT*,'SUMZY =          ',SUMZY
      LV=0
      DO 410 I=1,4
          LV=LV+1
          CC(I,3)=C(LV)
410   CONTINUE
      CALL DET(CC,DF,DN,IDF,ICC)
      PZSZ1=DN/DD
      READ (14,*) DELZ1
      PZZ1=(PZSZ1**2)*(DELZ1**2)
      PRINT*,'PZZ1=    ',PZZ1
      LV=4
      DO 420 I=1,4
          LV=LV+1
          CC(I,3)=C(LV)
420   CONTINUE
      CALL DET (CC,DF,DN,IDF,ICC)
      PZSZ2=DN/DD
      READ (14,*) DELZ2
      PZZ2=(PZSZ2**2)*(DELZ2**2)
      PRINT*,'PZZ2=    ',PZZ2
      LV=8
      DO 430 I=1,4
          LV=LV+1
          CC(I,3)=C(LV)
430   CONTINUE
      CALL DET (CC,DF,DN,IDF,ICC)
      PZSZ3=DN/DD
      READ (14,*) DELZ3
      PZZ3=(PZSZ3**2)*(DELZ3**2)
      PRINT*,'PZZ3=    ',PZZ3
      LV=12
      DO 440 I=1,4
          LV=LV+1
          CC(I,3)=C(LV)
440   CONTINUE
      CALL DET (CC,DF,DN,IDF,ICC)
      PZSZ4=DN/DD
      READ (14,*) DELZ4
      PZZ4=(PZSZ4**2)*(DELZ4**2)
      PRINT*,'PZZ4=   ',PZZ4
      SUMZZ=PZZ1+PZZ2+PZZ3+PZZ4
      PRINT*,'SUMZZ =          ',SUMZZ
      LV=0
      DO 450 I=1,4
          LV=LV+1
          CC(I,3)=E(LV)
450   CONTINUE
      CALL DET (CC,DF,DN,IDF,ICC)
      PZST1= DN/DD
      READ (14,*) DELT1
      PZT1=(PZST1**2)*(DELT1**2)
      PRINT*,'PZT1=     ',PZT1
      DO 460 I=1,4
```

```
      PZX3 = (PZSX3 **2) * (DELX3 **2)
      PRINT*,'PZX3 =          ',PZX3
      LV = 12
      DO 360 I=1,4
          LV = LV +1
          CC(I,3) = A(LV)
360   CONTINUE
      CALL DET (CC,DF,DN,IDF,ICC)
      PZSX4 = DN/DD
      READ (14,*) DELX4
      PZX4= (PZSX4**2) * (DELX4 **2)
      PRINT*,'PZX4=         ',PZX4
      SUMZX= PZX4 + PZX3 + PZX2 + PZX1
      PRINT*,'SUMZX=         ',SUMZX
      LV=0
      DO 370 I=1,4
          LV =LV+1
          CC(I,3)=B(LV)
370   CONTINUE
      CALL DET (CC,DF,DN,IDF,ICC)
      PZSY1= DN/DD
      READ (14,*) DELY1
      PZY1=(PZSY1**2)*(DELY1**2)
      PRINT*,'DELY1=      ',DELY1
      PRINT*,'PZY1=   ',PZY1
      LV=4
      DO 380 I=1,4
          LV=LV+1
          CC(I,3)=B(LV)
380   CONTINUE
      CALL DET (CC,DF,DN,IDF,ICC)
      PZSY2=DN/DD
      READ (14,*) DELY2
      PZY2=(PZSY2**2)*(DELY2**2)
      PRINT*,'PZY2=   ',PZY2
      LV=8
      DO 390 I=1,4
          LV=LV+1
          CC(I,3)=B(LV)
390   CONTINUE
      CALL DET (CC,DF,DN,IDF,ICC)
      PZSY3= DN/DD
      READ (14,*) DELY3
      PZY3=(PZSY3**2)*(DELY3**2)
      PRINT*,'PZY3=    ',PZY3
      LV=12
      DO 400 I=1,4
          LV=LV+1
          CC(I,3)=B(LV)
400   CONTINUE
      CALL DET (CC,DF,DN,IDF,ICC)
      PZSY4=DN/DD
      READ (14,*) DELY4
      PZY4=(PZSY4**2)*(DELY4**2)
      PRINT*,'PZY4=    ',PZY4
```

```
      PRINT*,'PYT2=  ',PYT2
      DO 310 I=1,4
          LV=LV+1
          BB(I,2)=E(LV)
310   CONTINUE
      CALL DET (BB,DF,DN,IDF,IBB)
      PYST3= DN/DD
      READ (14,*) DELT3
      PYT3=(PYST3**2)*(DELT3**2)
      PRINT*,'PYT3=    ',PYT3
      DO 320 I=1,4
          LV=LV+1
          BB(I,2)=E(LV)
320   CONTINUE
      CALL DET (BB,DF,DN,IDF,IBB)
      PYST4=DN/DD
      READ (14,*) DELT4
      PYT4=(PYST4**2)*(DELT4**2)
      PRINT*,'PYT4=    ',PYT4
      SUMYT=PYT1+PYT2+PYT3+PYT4
      PRINT*,'SUMYT =        ',SUMYT
      SUMY=SUMYX+SUMYY+SUMYZ+SUMYT
      PRINT*,' SUMY =       ',SUMY
      DELYS=SQRT(SUMY)
      PRINT*,'DELYS =        ',DELYS
      CALL MATCPY (DF,CC,N,M,IDF,ICC)
      LV = 0
      DO 330 I=1,4
          LV=LV+1
          CC(I,3) = A(LV)
330   CONTINUE
      CALL DET (CC,DF,DN,IDF,ICC)
      PZSX1 = DN/DD
      PRINT*,'PZSX1 =     ',PZSX1
      READ (14,*) DELX1
      PZX1 = (PZSX1 ** 2) * (DELX1 ** 2)
      PRINT*,'PZX1 =    ',PZX1
      LV=4
      DO 340 I=1,4
          LV = LV + 1
          CC(I,3) = A(LV)
340   CONTINUE
      CALL DET (CC,DF,DN,IDF,ICC)
      PZSX2 = DN/DD
      READ (14,*) DELX2
      PZX2 = (PZSX2**2)*(DELX2**2)
      PRINT*,'PZX2=     ',PZX2
      LV=8
      DO 350 I=1,4
          LV = LV + 1
          CC(I,3) = A(LV)
350   CONTINUE
      CALL DET (CC,DF,DN,IDF,ICC)
      PZSX3 = DN/DD
      READ (14,*) DELX3
```

```
      PYSZ1=DN/DD
      READ (14,*) DELZ1
      PYZ1=(PYSZ1**2)*(DELZ1**2)
      PRINT*,'DELZ1=    ',DELZ1
      PRINT*,'PYZ1=    ',PYZ1
      LV=4
      DO 260 I=1,4
          LV=LV+1
          BB(I,2)=C(LV)
260   CONTINUE
      CALL DET (BB,DF,DN,IDF,IBB)
      PYSZ2=DN/DD
      READ (14,*) DELZ2
      PYZ2=(PYSZ2**2)*(DELZ2**2)
      PRINT*,'PYZ2=    ',PYZ2
      LV=8
      DO 270 I=1,4
          LV=LV+1
          BB(I,2)=C(LV)
270   CONTINUE
      CALL DET (BB,DF,DN,IDF,IBB)
      PYSZ3=DN/DD
      READ (14,*) DELZ3
      PYZ3=(PYSZ3**2)*(DELZ3**2)
      PRINT*,'PYZ3=    ',PYZ3
      LV=12
      DO 280 I=1,4
          LV=LV+1
          BB(I,2)=C(LV)
280   CONTINUE
      CALL DET (BB,DF,DN,IDF,IBB)
      PYSZ4=DN/DD
      READ (14,*) DELZ4
      PYZ4=(PYSZ4**2)*(DELZ4**2)
      PRINT*,'PYZ4=  ',PYZ4
      SUMYZ=PYZ1+PYZ2+PYZ3+PYZ4
      PRINT*,'SUMYZ =         ',SUMYZ
      LV=0
      DO 290 I=1,4
          LV=LV+1
          BB(I,2)=E(LV)
290   CONTINUE
      CALL DET (BB,DF,DN,IDF,IBB)
      PYST1= DN/DD
      READ (14,*) DELT1
      PYT1=(PYST1**2)*(DELT1**2)
      PRINT*,'PYT1=    ',PYT1
      DO 300 I=1,4
          LV=LV+1
          BB(I,2)=E(LV)
300   CONTINUE
      CALL DET (BB,DF,DN,IDF,IBB)
      PYST2=DN/DD
      READ (14,*) DELT2
      PYT2=(PYST2**2)*(DELT2**2)
```

```
      CALL DET (BB,DF,DN,IDF,IBB)
      PYSX4 = DN/DD
      READ (14,*) DELX4
      PYX4= (PYSX4**2) * (DELX4 **2)
      PRINT*,´PYX4=       ´,PYX4
      SUMYX= PYX4 + PYX3 + PYX2 + PYX1
      PRINT*,´SUMYX=       ´,SUMYX
      LV=0
      DO 210 I=1,4
          LV =LV+1
          BB(I,2)=B(LV)
210   CONTINUE
      CALL DET (DF,DN,IDF,IBB)
      PYSY1= DN/DD
      READ (14,*) DELY1
      PYY1=(PYSY1**2)*(DELY1**2)
      PRINT*,´PYY1=  ´,PYY1
      LV=4
      DO 220 I=1,4
          LV=LV+1
          BB(I,2)=B(LV)
 220  CONTINUE
      CALL DET (BB,DF,DN,IDF,IBB)
      PYSY2=DN/DD
      READ (14,*) DELY2
      PYY2=(PYSY2**2)*(DELY2**2)
      PRINT*,´PYY2=  ´,PYY2
      LV=8
      DO 230 I=1,4
          LV=LV+1
          BB(I,2)=B(LV)
 230  CONTINUE
      CALL DET (BB,DF,DN,IDF,IBB)
      PYSY3= DN/DD
      READ (14,*) DELY3
      PYY3=(PYSY3**2)*(DELY3**2)
      PRINT*,´PYY3=   ´,PYY3
      LV=12
      DO 240 I=1,4
          LV=LV+1
          BB(I,2)=B(LV)
 240  CONTINUE
      CALL DET (BB,DF,DN,IDF,IBB)
      PYSY4=DN/DD
      READ (14,*) DELY4
      PYY4=(PYSY4**2)*(DELY4**2)
      PRINT*,´PYY4=   ´,PYY4
      SUMYY=PYY1+PYY2+PYY3+PYY4
      PRINT*,´SUMYY =        ´,SUMYY
      LV=0
      DO 250 I=1,4
          LV=LV+1
          BB(I,2)=C(LV)
 250  CONTINUE
      CALL DET(BB,DF,DN,IDF,IBB)
```

```
      PRINT*,'PXT3=    ',PXT3
      DO 160 I=1,4
          LV=LV+1
          AA(I,1)=E(LV)
160   CONTINUE
      CALL DET (AA,DF,DN,IDF,IAA)
      PXST4=DN/DD
      READ (14,*) DELT4
      PXT4=(PXST4**2)*(DELT4**2)
      PRINT*,'PXT4=    ',PXT4
      SUMXT=PXT1+PXT2+PXT3+PXT4
      PRINT*,'SUMXT =       ',SUMXT
      SUMX=SUMXX+SUMXY+SUMXZ+SUMXT
      PRINT*,' SUMX =      ',SUMX
      DELXS=SQRT(SUMX)
      PRINT*,'DELXS =       ',DELXS
      CALL MATCPY (DF,BB,N,M,IDF,IBB)
      LV =0
      DO 170 I=1,4
          LV = LV + 1
          BB(I,2) = A(LV)
170   CONTINUE
      CALL DET(BB,DF,DN,IDF,IBB)
      CALL MATPRT (BB,N,M,IBB)
      PYSX1 =  DN/DD
      PRINT*,'PYSX1=    ',PYSX1
      READ (14,*)  DELX1
      PRINT*,'DELX1=   ',DELX1
      PYX1 = (PYSX1 ** 2) * (DELX1 ** 2)
      PRINT*,'PYX1 =    ',PYX1
      LV=4
      DO 180 I=1,4
          LV = LV + 1
          BB(I,2) = A(LV)
180   CONTINUE
      CALL DET (BB,DF,DN,IDF,IBB)
      PYSX2 = DN/DD
      READ (14,*) DELX2
      PYX2 = (PYSX2**2)*(DELX2**2)
      PRINT*,'PYX2=     ',PYX2
      LV=3
      DO 190 I=1,4
          LV = LV + 1
          BB(I,2) = A(LV)
190   CONTINUE
      CALL DET (BB,DF,DN,IDF,IBB)
      PYSX3 = DN/DD
      READ (14,*) DELX3
      PYX3 = (PYSX3 **2) * (DELX3 **2)
      PRINT*,'PYX3 =        ',PYX3
      LV = 12
      DO 200 I=1,4
          LV = LV +1
          BB(I,2) = A(LV)
200   CONTINUE
```

```
      LV=12
      DO 120 I=1,4
          LV=LV+1
          AA(I,1)=C(LV)
120   CONTINUE
      CALL DET (AA,DF,DN,IDF,IAA)
      PXSZ4=DN/DD
      READ (14,*) DELZ4
      PXZ4=(PXSZ4**2)*(DELZ4**2)
      PRINT*,'PXZ4=  ',PXZ4
      SUMXZ=PXZ1+PXZ2+PXZ3+PXZ4
      PRINT*,'SUMXZ =        ',SUMXZ
      E(1)=-EONE
      E(2)=0
      E(3)=0
      E(4)=0
      E(5)=0
      E(6)=-ETWO
      E(7)=0
      E(8)=0
      E(9)=0
      E(10)=0
      E(11)=-ETHREE
      E(12)=0
      E(13)=0
      E(14)=0
      E(15)=0
      E(16)=-EFOUR
      LV=0
      DO 130 I=1,4
          LV=LV+1
          AA(I,1)=E(LV)
130   CONTINUE
      CALL DET (AA,DF,DN,IDF,IAA)
      PXST1= DN/DD
      READ (14,*) DELT1
      PXT1=(PXST1**2)*(DELT1**2)
      PRINT*,'PXT1=    ',PXT1
      DO 140 I=1,4
          LV=LV+1
          AA(I,1)=E(LV)
140   CONTINUE
      CALL DET (AA,DF,DN,IDF,IAA)
      PXST2=DN/DD
      READ (14,*) DELT2
      PXT2=(PXST2**2)*(DELT2**2)
      PRINT*,'PXT2=  ',PXT2
      DO 150 I=1,4
          LV=LV+1
          AA(I,1)=E(LV)
150   CONTINUE
      CALL DET (AA,DF,DN,IDF,IAA)
      PXST3= DN/DD
      READ (14,*) DELT3
      PXT3=(PXST3**2)*(DELT3**2)
```

```
         AA(I,1)=B(LV)
 80   CONTINUE
      CALL DET (AA,DF,DN,IDF,IAA)
      PXSY4=DN/DD
      READ (14,*) DELY4
      PXY4=(PXSY4**2)*(DELY4**2)
      PRINT*,´PXY4=   ´,PXY4
      SUMXY=PXY1+PXY2+PXY3+PXY4
      PRINT*,´SUMXY =         ´,SUMXY
      C(1)=-CONE
      C(2)=0
      C(3)=0
      C(4)=0
      C(5)=0
      C(6)=-CTWO
      C(7)=0
      C(8)=0
      C(9)=0
      C(10)=0
      C(11)=-CTHREE
      C(12)=0
      C(13)=0
      C(14)=0
      C(15)=0
      C(16)=-CFOUR
      LV=0
      DO 90 I=1,4
         LV=LV+1
         AA(I,1)=C(LV)
 90   CONTINUE
      CALL DET(AA,DF,DN,IDF,IAA)
      PXSZ1=DN/DD
      READ (14,*) DELZ1
      PXZ1=(PXSZ1**2)*(DELZ1**2)
      PRINT*,´PXZ1=   ´,PXZ1
      LV=4
      DO 100 I=1,4
         LV=LV+1
         AA(I,1)=C(LV)
 100  CONTINUE
      CALL DET (AA,DF,DN,IDF,IAA)
      PXSZ2=DN/DD
      READ (14,*) DELZ2
      PXZ2=(PXSZ2**2)*(DELZ2**2)
      PRINT*,´PXZ2=   ´,PXZ2
      LV=8
      DO 110 I=1,4
         LV=LV+1
         AA(I,1)=C(LV)
 110  CONTINUE
      CALL DET (AA,DF,DN,IDF,IAA)
      PXSZ3=DN/DD
      READ (14,*) DELZ3
      PXZ3=(PXSZ3**2)*(DELZ3**2)
      PRINT*,´PXZ3=   ´,PXZ3
```

```
      PXSX4 = DN/DD
      READ (14,*) DELX4
      PXX4= (PXSX4**2) * (DELX4 **2)
      PRINT*,'PXX4=        ',PXX4
      SUMXX= PXX4 + PXX3 + PXX2 + PXX1
      PRINT*,'SUMXX=       ',SUMXX
      B(1) = -BONE
      B(2) = 0
      B(3) = 0
      B(4) = 0
      B(5) = 0
      B(6) = -BTWO
      B(7 ) = 0
      B(8 ) = 0
      B(9 ) = 0
      B(10 ) = 0
      B(11) = -BTHREE
      B(12) = 0
      B(13) = 0
      B(14) = 0
      B(15) = 0
      B(16) = -BFOUR
      LV=0
      DO 50 I=1,4
           LV =LV+1
           AA(I,1)=B(LV)
50    CONTINUE
      CALL DET (AA,DF,DN,IDF,IAA)
      PXSY1= DN/DD
      READ (14,*) DELY1
      PXY1=(PXSY1**2)*(DELY1**2)
      PRINT*,'PXY1=  ',PXY1
      LV=4
      DO 60 I=1,4
          LV=LV+1
          AA(I,1)=B(LV)
60    CONTINUE
      CALL DET (AA,DF,DN,IDF,IAA)
      PXSY2=DN/DD
      READ (14,*) DELY2
      PXY2=(PXSY2**2)*(DELY2**2)
      PRINT*,'PXY2=  ',PXY2
      LV=8
      DO 70 I=1,4
          LV=LV+1
          AA(I,1)=B(LV)
70    CONTINUE
      CALL DET (AA,DF,DN,IDF,IAA)
      PXSY3= DN/DD
      READ (14,*) DELY3
      PXY3=(PXSY3**2)*(DELY3**2)
      PRINT*,'PXY3=   ',PXY3
      LV=12
      DO 80 I=1,4
          LV=LV+1
```

```
      A(5) = 0
      A(6) = -ATWO
      A(7) = 0
      A(8) = 0
      A(9) = 0
      A(10) = 0
      A(11) = -ATHREE
      A(12) = 0
      A(13) = 0
      A(14) = 0
      A(15) = 0
      A(16) = -AFOUR
      LV =0
      DO 10 I=1,4
          LV = LV + 1
          AA(I,1) = A(LV)
10    CONTINUE
      CALL MATPRT (AA,N,M,IA)
      CALL DET(AA,DF,DN,IDF,IAA)
      PXSX1 =  DN/DD
      PRINT*,'PXSX1=    ',PXSX1
      READ (14,*)  DELX1
      PRINT*,'DELX1=    ',DELX1
      PXX1 = (PXSX1 ** 2) * (DELX1 ** 2)
      PRINT*,'PXX1 =    ',PXX1
      CALL MATCPY (DF,AA,N,M,IDF,IAA)
      CALL MATPRT (AA,N,M,IAA)
      LV=4
      DO 20 I=1,4
          LV = LV + 1
          AA(I,1) = A(LV)
20    CONTINUE
      CALL MATPRT (AA,N,M,IAA)
      CALL DET (AA,DF,DN,IDF,IAA)
      PRINT*,'DN=      ',DN
      PXSX2 = DN/DD
      READ (14,*) DELX2
      PXX2 = (PXSX2**2)*(DELX2**2)
      PRINT*,'PXX2=     ',PXX2
      LV=8
      DO 30 I=1,4
          LV = LV + 1
          AA(I,1) = A(LV)
30    CONTINUE
      CALL DET (AA,DF,DN,IDF,IAA)
      PXSX3 = DN/DD
      READ (14,*) DELX3
      PXX3 = (PXSX3 **2) * (DELX3 **2)
      PRINT*,'PXX3 =         ',PXX3
      LV = 12
      DO 40 I=1,4
          LV = LV +1
          AA(I,1) = A(LV)
40    CONTINUE
      CALL DET (AA,DF,DN,IDF,IAA)
```

```
BTHREE = 2*(YS - Y3)
PRINT*,'BTHREE=   ',BTHREE
BFOUR = 2*(YS - Y4)
PRINT*,'BFOUR=    ',BFOUR
READ  (15,*) ZS,Z1,Z2,Z3,Z4
CONE = 2*(ZS - Z1)
PRINT*, 'CONE=    ',CONE
CTWO = 2*(ZS - Z2)
PRINT*,'CTWO=     ',CTWO
CTHREE = 2*(ZS - Z3)
PRINT*,'CTHREE=     ',CTHREE
CFOUR = 2*(ZS - Z4)
PRINT*,'CFOUR=    ',CFOUR
READ  (15,*) TS,T1,T2,T3,T4
EONE = 2*(T1 - TS)
PRINT*,'EONE =        ',EONE
ETWO = 2*(T2 - TS)
PRINT*,'ETWO=   ',ETWO
ETHREE = 2*(T3 - TS)
PRINT*,'ETHREE=   ',ETHREE
EFOUR = 2*(T4 - TS)
PRINT*,'EFOUR=   ',EFOUR
D(1,1) = AONE
D(2,1) = ATWO
D(3,1) = ATHREE
D(4,1) = AFOUR
D(1,2) = BONE
D(2,2) = BTWO
D(3,2) = BTHREE
D(4,2) = BFOUR
D(1,3) = CONE
D(2,3) = CTWO
D(3,3) = CTHREE
D(4,3) = CFOUR
D(1,4) = EONE
D(2,4) = ETWO
D(3,4) = ETHREE
D(4,4) = EFOUR
PRINT*,'D'
CALL MATPRT(D,N,M,ID)
CALL MATCPY (D,AA,N,M,ID,IAA)
PRINT*,'AA'
CALL MATPRT (AA,N,M,IAA)
CALL MATCPY (D,DF,N,M,ID,IDF)
PRINT*,'DF'
CALL MATPRT (DF,N,M,IDF)
IJOB = 4
D1 = 1
CALL LINV3F (D,IJOB,IJOB,N,ID,D1,D2,WKAREA,IER)
DD = D1*(2**D2)
PRINT *,'DD =      ',DD
A(1) = -AONE
A(2) = 0
A(3) = 0
A(4) = 0
```

```
*
*************************************************************

      PROGRAM Q

      INTEGER IJOB,IER,N,ID,IAA,IBB,ICC,ITT,M,I,LV,IDF

      REAL AONE,ATWO,ATHREE,AFOUR,BONE,BTWO,BTHREE,BFOUR,
     CCONE,CTWO,CTHREE,CFOUR,EONE,ETWO,ETHREE,EFOUR,X1,X2,X3,
     CX4,Y1,Y2,Y3,Y4,Z1,Z2,Z3,Z4,T1,T2,T3,T4,XS,YS,ZS,TS,
     CDELX1,DELX2,DELX3,DELX4,DELY1,DELY2,DELY3,DELY4,DELZ1,
     CDELZ2,DELZ3,DELZ4,DELT1,DELT2,DELT3,DELT4,DELXS,DELYS,
     CDELZS,DELTS,DN,SUMXX,SUMXY,SUMXZ,SUMXT,SUMX,A,B,C,D,
     CWKAREA,DF,PXSX1,PXSX2,PXSX3,PXSX4,PXSY1,PXSY2,PXSY3,
     CPXSY4,PXSZ1,PXSZ2,PXSZ3,PXSZ4,PXST1,PXST2,PXST3,PXST4,
     CPXX1,PXX2,PXX3,PXX4,PXY1,PXY2,PXY3,PXY4,PXZ1,PXZ2,PXZ3,
     CPXZ4,PXT1,PXT2,PXT3,PXT4,PYSX1,PYSX2,PYSX3,PYSX4,PYSY1,
     CPYSY2,PYSY3,PYSY4,PYSZ1,PYSZ2,PYSZ3,PYSZ4,PYST1,PYST2,
     CPYST3,PYST4,SUMYX,SUMYY,SUMYZ,SUMYT,SUMY,PZSX1,PZSX2,
     CPZSX3,PZSX4,PZSY1,PZSY2,PZSY3,PZSY4,PZSZ1,PZSZ2,PZSZ3,
     CPZSZ4,PZST1,PZST2,PZST3,PZST4,SUMZX,SUMZY,SUMZZ,SUMZT,
     CSUMZ,PTSX1,PTSX2,PTSX3,PTSX4,PTSY1,PTSY2,PTSY3,PTSY4,
     CPTSZ1,PTSZ2,PTSZ3,PTSZ4,PTST1,PTST2,PTST3,PTST4,SUMTX,
     CSUMTY,SUMTZ,SUMTT,SUMT,PYX1,PYX2,PYX3,PYX4,PYY1,PYY2,
     CPYY3,PYY4,PYZ1,PYZ2,SUMY,SUMZ,SUMT,PYZ3,PYZ4,PYT1,PYT2,
     CPYT3,PYT4,PZX1,PZX2,PZX3,PZX4,PZT1,PZT2,PZY1,PZY2,PZY3,
     CPZY4,PZZ1,PZZ2,PZZ3,PZZ4,PTZ1,PTZ2,PTZ3,PTZ4,PZT4,PTX1,
     CPTX2,PTX3,PTX4,PTY1,PTY2,PTY3,PTY4,PTT1,PTT2,PTT3,PTT4,
     CAA,BB,CC,TT

      DIMENSION A(16),B(16),C(16),AA(4,4),D(4,4),E(16),
     CWKAREA(3),DF(4,4),BB(4,4),CC(4,4),TT(4,4)

      PARAMETER (ID=4,IAA=4,IBB=4,ICC=4,IDD=4,IDF=4)

      OPEN (14, FILE ='DEL')
      OPEN (15, FILE = 'DT1')
      READ (15,*) XS,X1,X2,X3,X4
      N=4
      M=4
      PRINT*,'READ X'
      AONE=2*(XS - X1)
      PRINT*,'AONE=     ',AONE
      ATWO=2*(XS - X2)
      PRINT*,'ATWO =    ',ATWO
      ATHREE= 2*(XS - X3)
      PRINT*,'ATHREE=   ',ATHREE
      AFOUR= 2*(XS - X4)
      PRINT*,'AFOUR=     ',AFOUR
      READ  (15,*) YS,Y1,Y2,Y3,Y4
      PRINT*,'READ Y'
      BONE = 2*(YS -Y1)
      PRINT*,'BONE=    ',BONE
      BTWO = 2*(YS - Y2)
      PRINT*,'BTWO=    ',BTWO
```

```
*                                DER. AND THE DELX1-X4
* PZY1-Y4        REAL         PRODUCT OF THE SQUARES OF PART.
*                                DER. AND THE DELY1-Y4
* PZZ1-Z4        REAL         PRODUCT OF THE SQUARES OF PART.
*                                DER. AND THE DELZ1-Z4
* PZT1-T4        REAL         PRODUCT OF THE SQUARES OF PART.
*                                DER. AND THE DELT1-T4
* SUMZX          REAL         SUM OF PRODUCTS INVOLVING DELXI'S
*                                AND PART. DER. OF ZS
* SUMZY          REAL         SUM OF PRODUCTS INVOLVING DELYI'S
*                                AND PART. DER. OF ZS
* SUMZZ          REAL         SUM OF PRODUCTS INVOLVING DEELZI'S
*                                AND PART. DER. OF ZS
* SUMZT          REAL         SUM OF PRODUCTS INVOLVING DELTI'S
*                                AND PART. DER. OF ZS
* SUMZ           REAL         SUM OF ALL PRODUCTS INVOLVING PART.
*                                DER. OF ZS
* PTSX1-X4       REAL         PART. DER. OF TS WRT X1-X4
* PTSY1-Y4       REAL         PART. DER. OF TS WRT Y1-Y4
* PTSZ1-Z4       REAL         PART. DER. OF TS WRT Z1-Z4
* PTST1-T4       REAL         PART. DER. OF TS WRT T1-T4
* PTX1-X4        REAL         PRODUCT OF THE SQUARES OF PART.
*                                DER. AND THE DELX1-X4
* PTY1-Y4        REAL         PRODUCT OF THE SQUARES OF PART.
*                                DER. AND THE DELY1-Y4
* PTZ1-Z4        REAL         PRODUCT OF THE SQUARES OF PART.
*                                DER. AND THE DELZ1-Z4
* PTT1-T4        REAL         PRODUCT OF THE SQUARES OF PART.
*                                DER. AND THE DELT1-T4
* SUMTX          REAL         SUM OF PRODUCTS INVOLVING DELXI'S
*                                AND PART. DER. OF TS
* SUMTY          REAL         SUM OF PRODUCTS INVOLVING DELYI'S
*                                AND PART. DER. OF TS
* SUMTZ          REAL         SUM OF PRODUCTS INVOLVING DELZI'S
*                                AND PART. DER. OF TS
* SUMTT          REAL         SUM OF PRODUCTS INVOLVING DELTI'S
*                                AND PART. DER. OF TS
* SUMT           REAL         SUM OF ALL PRODUCTS INVOLVING PART.
*                                DER. OF TS
************************   NOTE   ************************
*
*    PART. DER. = PARTIAL DERIVATIVE
*    WRT        = WITH RESPECT TO
*
**********************************************************
*
* MODULES        ARGUEMENTS          PURPOSE
* CALLED:        PASSED:
*
* LINV3F         A,B,IJOB,N,IA,D1,   FIND THE DETERMINANT
*                 D2,WKAREA,IER
* DET            A,B,DN,IB,IA        SOLVES FOR THE JACOBIAN
*                                     MATRIX
* MATPRT         A,N,M,IA            PRINT MATRIX
* MATCPY         A,C,N,M,IA,IC       COPY MATRIX
```

```
      DO 510 I=1,4
          LV = LV + 1
          TT(I,4) = A(LV)
510   CONTINUE
      CALL DET (TT,DF,DN,IDF,ITT)
      PTSX3 = DN/DD
      READ (14,*) DELX3
      PTX3 = (PTSX3 **2) * (DELX3 **2)
      PRINT*,´PTX3 =         ´,PTX3
      LV = 12
      DO 520 I=1,4
          LV = LV +1
          TT(I,4) = A(LV)
520   CONTINUE
      CALL DET (TT,DF,DN,IDF,ITT)
      PTSX4 = DN/DD
      READ (14,*) DELX4
      PTX4= (PTSX4**2) * (DELX4 **2)
      PRINT*,´PTX4=        ´,PTX4
      SUMTX= PTX4 + PTX3 + PTX2 + PTX1
      PRINT*,´SUMTX=         ´,SUMTX
      LV=0
      DO 530 I=1,4
          LV = LV+1
          TT(I,4)=B(LV)
530   CONTINUE
      CALL DET (TT,DF,DN,IDF,ITT)
      PTSY1= DN/DD
      READ (14,*) DELY1
      PTY1=(PTSY1**2)*(DELY1**2)
      PRINT*,´DELY1=      ´,DELY1
      PRINT*,´PTY1=   ´,PTY1
      LV=4
      DO 540 I=1,4
          LV=LV+1
          TT(I,4)=B(LV)
540   CONTINUE
      CALL DET (TT,DF,DN,IDF,ITT)
      PTSY2=DN/DD
      READ (14,*) DELY2
      PTY2=(PTSY2**2)*(DELY2**2)
      PRINT*,´PTY2=   ´,PTY2
      LV=8
      DO 550 I=1,4
          LV=LV+1
          TT(I,4)=B(LV)
550   CONTINUE
      CALL DET (TT,DF,DN,IDF,ITT)
      PTSY3= DN/DD
      READ (14,*) DELY3
      PTY3=(PTSY3**2)*(DELY3**2)
      PRINT*,´PTY3=    ´,PTY3
      LV=12
      DO 560 I=1,4
          LV=LV+1
```

```
          TT(I,4)=B(LV)
  560   CONTINUE
        CALL DET (TT,DF,DN,IDF,ITT)
        PTSY4=DN/DD
        READ (14,*) DELY4
        PTY4=(PTSY4**2)*(DELY4**2)
        PRINT*,'PTY4=    ',PTY4
        SUMTY=PTY1+PTY2+PTY3+PTY4
        PRINT*,'SUMTY =         ',SUMTY
        LV=0
        DO 570 I=1,4
          LV=LV+1
          TT(I,4)=C(LV)
  570   CONTINUE
        CALL DET(TT,DF,DN,IDF,ITT)
        PTSZ1=DN/DD
        READ (14,*) DELZ1
        PTZ1=(PTSZ1**2)*(DELZ1**2)
        PRINT*,'PTZ1=    ',PTZ1
        LV=4
        DO 580 I=1,4
          LV=LV+1
          TT(I,4)=C(LV)
  580   CONTINUE
        CALL DET (TT,DF,DN,IDF,ITT)
        PTSZ2=DN/DD
        READ (14,*) DELZ2
        PTZ2=(PTSZ2**2)*(DELZ2**2)
        PRINT*,'PTZ2=    ',PTZ2
        LV=8
        DO 590 I=1,4
          LV=LV+1
          TT(I,4)=C(LV)
  590   CONTINUE
        CALL DET (TT,DF,DN,IDF,ITT)
        PTSZ3=DN/DD
        READ (14,*) DELZ3
        PTZ3=(PTSZ3**2)*(DELZ3**2)
        PRINT*,'PTZ3=    ',PTZ3
        LV=12
        DO 600 I=1,4
          LV=LV+1
          TT(I,4)=C(LV)
  600   CONTINUE
        CALL DET (TT,DF,DN,IDF,ITT)
        PTSZ4=DN/DD
        READ (14,*) DELZ4
        PTZ4=(PTSZ4**2)*(DELZ4**2)
        PRINT*,'PTZ4=   ',PTZ4
        SUMTZ=PTZ1+PTZ2+PTZ3+PTZ4
        PRINT*,'SUMTZ =         ',SUMTZ
        LV=0
        DO 610 I=1,4
          LV=LV+1
          TT(I,4)=E(LV)
```

```
610   CONTINUE
      CALL DET (TT,DF,DN,IDF,ITT)
      PTST1= DN/DD
      READ (14,*) DELT1
      PTT1=(PTST1**2)*(DELT1**2)
      PRINT*,'PTT1=    ',PTT1
      DO 620 I=1,4
          LV=LV+1
          TT(I,4)=E(LV)
620   CONTINUE
      CALL DET (TT,DF,DN,IDF,ITT)
      PTST2=DN/DD
      READ (14,*) DELT2
      PTT2=(PTST2**2)*(DELT2**2)
      PRINT*,'PTT2=  ',PTT2
      DO 630 I=1,4
          LV=LV+1
          TT(I,4)=E(LV)
630   CONTINUE
      CALL DET (TT,DF,DN,IDF,ITT)
      PTST3= DN/DD
      READ (14,*) DELT3
      PTT3=(PTST3**2)*(DELT3**2)
      PRINT*,'PTT3=    ',PTT3
      DO 640 I=1,4
          LV=LV+1
          TT(I,4)=E(LV)
640   CONTINUE
      CALL DET (TT,DF,DN,IDF,ITT)
      PTST4=DN/DD
      READ (14,*) DELT4
      PTT4=(PTST4**2)*(DELT4**2)
      PRINT*,'PTT4=    ',PTT4
      SUMTT=PTT1+PTT2+PTT3+PTT4
      PRINT*,'SUMTT =       ',SUMTT
      SUMT=SUMTX+SUMTY+SUMTZ+SUMTT
      PRINT*,' SUMT =       ',SUMT
      DELTS=SQRT(SUMT)
      PRINT*,'DELTS =       ',DELTS
      END
```

```
************************************************************
*
* SUBROUTINE NAME: MATCPY
*
* ARGUEMENT LIST: A,C,N,M,IA,IC
* CALLED BY:   Q
* PROJECT:  THESIS                    DATE: 15 OCT 84
* PROGRAMMER: C. M. WOZNIAKOWSKI
*
************************************************************
*
* MODULE DESCRIPTION: COPIES (STORES) A MATRIX
*
************************************************************
*
* ARGUEMENT     IN/  TYPE  PASSED/  PURPOSE
* NAME          OUT        GLOBAL
*
* A(IA,M)       IN   REAL  PASSED   MATRIX TO BE COPIED
* C(IC,M)       OUT  REAL  PASSED   COPIED MATRIX
* N             IN   INT   PASSED   ROW DIMENSION
* M             IN   INT   PASSED   COL DIMENSION
* IA            IN   INT   PASSED   MAX ROW DIMENSION FOR A
* IC            IN   INT   PASSED   MAX ROW DIMENSION FOR C
*
************************************************************
*
* DESCRIPTION OF ALGORITHM DEVELOPMENT:
*
* FOR (I=1,N) LOOP
*    FOR (J=1,M) LOOP
*         C(I,J) = A(I,J)
*    END LOOP
* END LOOP
* END
*
************************************************************
*
* LOCAL VARIABLES   TYPE      PURPOSE
*
*    I,J            INT       COUNTING VARIABLES
*
************************************************************

      SUBROUTINE MATCPY (A,C,N,M,IA,IC)

      INTEGER N,M,IA,IC,I,J
      REAL A,C
      DIMENSION A(4,4),C(4,4)

      DO 56 I=1,N
          DO 55 J=1,M
               C(I,J) = A(I,J)
 55         CONTINUE
 56   CONTINUE
```

```
*************************************************************
*
* SUBROUTINE NAME: MATCPY
*
* ARGUEMENT LIST: A,C,N,M,IA,IC
* CALLED BY:   Q
* PROJECT:  THESIS                    DATE: 15 OCT 84
* PROGRAMMER: C. M. WOZNIAKOWSKI
*
*************************************************************
*
* MODULE DESCRIPTION: COPIES (STORES) A MATRIX
*
*************************************************************
*
* ARGUEMENT     IN/  TYPE  PASSED/  PURPOSE
* NAME          OUT        GLOBAL
*
* A(IA,M)       IN   REAL  PASSED   MATRIX TO BE COPIED
* C(IC,M)       OUT  REAL  PASSED   COPIED MATRIX
* N             IN   INT   PASSED   ROW DIMENSION
* M             IN   INT   PASSED   COL DIMENSION
* IA            IN   INT   PASSED   MAX ROW DIMENSION FOR A
* IC            IN   INT   PASSED   MAX ROW DIMENSION FOR C
*
*************************************************************
*
* DESCRIPTION OF ALGORITHM DEVELOPMENT:
*
* FOR (I=1,N) LOOP
*    FOR (J=1,M) LOOP
*         C(I,J) = A(I,J)
*    END LOOP
* END LOOP
* END
*
*************************************************************
*
* LOCAL VARIABLES   TYPE      PURPOSE
*
*    I,J            INT       COUNTING VARIABLES
*
*************************************************************

      SUBROUTINE MATCPY (A,C,N,M,IA,IC)

      INTEGER N,M,IA,IC,I,J
      REAL A,C
      DIMENSION A(4,4),C(4,4)

      DO 56 I=1,N
          DO 55 J=1,M
               C(I,J) = A(I,J)
 55        CONTINUE
 56   CONTINUE
```

END

```
************************************************************
*
* SUBROUTINE NAME:  MATPRT
*
* ARGUEMENT LIST:   A,N,M,IA
* CALLED BY:   Q
* PROJECT: THESIS                    DATE:  15 OCT 84
* PROGRAMMER:  C. M. WOZNIAKOWSKI
*
************************************************************
*
* MODULE DESCRIPTION: PRINT A MATRIX
*
************************************************************
*
* ARGUEMENT    IN/  TYPE  PASSED/  PURPOSE
* NAME         OUT        GLOBAL
*
* A(N,M)       I/O  REAL  PASSED   THE MATRIX ELEMENTS
* N            IN   INT   PASSED   ROWS IN MATRIX
* M            IN   INT   PASSED   COLUMNS IN MATRIX
* IA           IN   INT   PASSED   MAX ROW DIMENSION
* STD OUTPUT   OUT  TEXT  GLOBAL   OUTPUT MATRIX
*
************************************************************
*
* DESCRIPTION OF ALGORITHM DEVELOPMENT:
*
* PRINT BLANK LINES
* FOR I=1,N LOOP
*    FOR J=1,M LOOP
*         PRINT (FORMAT) MATRIX A (I,J)
*         FORMAT (5E19.7)
*    END LOOP
* END LOOP
* END
*
************************************************************
*
* LOCAL VARIABLES   TYPE      PURPOSE
*
*    I,J            INT       COUNTING VARIABLES
*
************************************************************

      SUBROUTINE MATPRT (A,N,M,IA)

      DIMENSION A(4,4)
      REAL A

      PRINT *
      DO 333 I=1,N
          PRINT *
          PRINT 444, (A(I,J),J=1,M)
```

```
444      FORMAT (5E17.7)
333   CONTINUE
      END
```

```
************************************************************
*
* SUBROUTINE NAME:  DET
*
* ARGUEMENT LIST:   A,C,DN,IC,IA
* CALLED BY:  Q
*
************************************************************
*
*  MODULE DESCRIPTION:  FINDS THE DETERMINANT AND RECOPIES A
*    MATRIX FOR LATER USE
*
************************************************************
*
* ARGUEMENT     IN/  TYPE  PASSED/  PURPOSE
* NAME          OUT        GLOBAL
*
* A             I/O  REAL  PASSED   MATRIX OPERATED ON
* C             IN   REAL  PASSED   MATRIX TO BE COPIED
* DN            OUT  REAL  PASSED   THE DETERMINANT
* IC            IN   INT   PASSED   MAX ROW DIMENSION FOR C
* IA            IN   INT   PASSED   MAX ROW DIMENSION FOR A
*
************************************************************
*
* DESCRIPTION OF ALGORITHM DEVELOPMENT:
*
* SET VARIABLES TO A CONSTANT
* FIND THE DETERMINANT USING THE IMSL SUBROUTINE LINV3F
* COPIES ORGINAL MATRIX
* DN = D1 * (2 ** D2)
* END
*
************************************************************
*
* LOCAL VARABLES     TYPE  PURPOSE
*
* IJOB               INT   INPUT OPTION PARAMETER
* D1                 INT   I/O,  IF D1 AND D2 COMPONENTS  OF
*                          DETERMINANT DESIRED, INPUT D1 > 0
* IA                 INT   MAX ROW DIMENSION FOR A
* N                  INT   ROWS IN MATRIX A
* M                  INT   COLUMNS IN MATRIX A
*
************************************************************
*
* MODULES      ARGUEMENT           PUPOSE
* CALLED       PASSED
*
* LINV3F       (A,B,IJOB,N,IA,D1,  FIND THE DETERMINANT
*               D2,WKAREA,IER)
* MATCPY       (A,C,N,M,IA,IC)     COPY MATRIX
*
************************************************************
```

```
SUBROUTINE DET (A,C,DN,IC,IA)

REAL DF(4,4),TT(4,4),WKAREA(8),D1,D2,DN
INTEGER N,M,IDF,ITT,IER,IJOB

IJOB=4
D1=4
ITT=4
N=4
M=4
CALL LINV3F (A,IJOB,IJOB,N,IA,D1,D2,WKAREA,IER)
CALL MATCPY (C,A,N,M,IC,IA)
DN = D1 * (2 ** D2)
END
```

BIBLIOGRAPHY

1. Baird, David C. Probability and Experimental Errors in Science. Englewood Cliffs, NJ: Prentice-Hall, 1962.

2. Connor, J. P. et al. "The Recent Appearance of a New X-Ray Source in the Southern Sky," The Astrophysical Journal, 157: L157-L159 (September 1969).

3. Coloquitt, Elodie S. Beacon Tracking Systems Simulation Naval Surface Weapons Center, Dahlgreen VA, June 1976 (AD-A038 643).

4. Denaro, Robert P. "NAVSTAR: The All-purpose Satellites," IEEE Spectrum, 18: 35-40 (May 1981).

5. "Design Review of NAVSTAR Block 2 Completed," Aviation Week and Space Technology, 116: 88-91 (June 7, 1982).

6. Elson, Benjamin M. "Transoceanic Flight Shows GPS Uses," Aviation Week and Space Technology, 118: 45-48 (July 25, 1983).

7. Euler, William C. and James W. Breedlove. "NAVSTAR-Global Positioning System - A Revolutionary Capability," IEEE Electronics and Aerospace Systems Conventions. 217-223. IEEE Publications, Washington, D. C., November 1981.

8. Glasstone, Samuel. The Effects of Nuclear Weapons. Washington D. C.: United States Atomic Energy Commission, April 1962.

9. Hones, E. W. Jr., I. D. Palmer and P. R. Higbie. "Energetic Protons of Magnetospheric Origin in the Plasma Sheet Associated With Substorms," Journal of Geophysical Research, 81: 3866-3874 (August 1, 1976).

10. Hones, E. W. Jr., D. N. Baker and W. C. Feldman. Evaluation of Some Geophysical Events on 22 September 1979 Los Alamos Scientific Lab., NM, April 1981 (LA-8672).

11. IMSL Library Reference# Manual (Edition# 9) Houston, Tx: International Mathematical and Statistical Libraries, Inc., 1982.

12. Klauss, Philip J. "Clandestine Nuclear Test Doubted," Aviation Week and Space Technology, 99: 67-72 (August 11,1980).

13. -----, "Compromise Reached on NAVSAT," Aviation Week and Space Technology, 99: 46-51 (November 26, 1973).

14. Klebesadel, Ray W. et al. "Observations of Gamma-Ray Bursts of Cosmic Origin," The Astrophysical Journal, 182: L85-L88 (June 1, 1973).

15. Klepczynski, William J. "Modern Navigation Systems and their Relations to Timekeeping," Proceedings of IEEE, 71: 1193-1198 (October, 1983).

16. LaBahn, R. W. and A. K. Paul. HF Propagation Modes for 5 and 15 MHz over a 1400km Midlatitude Path. Naval Ocean Systems Center, San Diego CA, October 1983 (AD-B078 350).

17. Marshall, Joe. Chief, Evaluation Branch, Space Division. Personnel correspondence. HQ AFTAC, Patrick AFB FL, June 1984.

18. Miller, Barry. "Satellite Navigation Network Defined," Aviation Week and Space Technology, 100: 22-23 (April 15, 1974).

19. Milliken, R. J. and C. J. Zoller. "Principle of Operation of NAVSTAR and System Characteristics," Global Positioning System, Papers published in Navigation, 25: 3-14 (Spring, 1978).

20. Mohan, S. N. "Ship Navigation using NAVSTAR GPS: An Application Study," The Journal of the Astronautical Sciences, 32: 81-92 (January-March 1984).

21. Parkinson, Bradford W. and Stephen W. Gilbert. "NAVSTAR: Global Positioning System - Ten Years Later," Proceedings of IEEE, 71: 1177-1186 (October, 1983).

22. Prilutskiy, O. F. et al. "Powerful Gamma Ray Splashes, A New Astronomical# Discovery," Priroda, No. 3 (703): 93-95 (March 1974) (NASA-TT-F-15789).

23. Proctor, D. E. "A Hyperbolic System for Obtaining VHF Radio Pictures of Lighting," Journal of Geophysical Research, 76: 1478-1489 (February 20, 1971).

24. Richards, Geoff "NAVSTAR - A Complete Global Navigation System," Spaceflight, 22: 2-6 (January, 1980).

25. Rustan, P. L. and et. al. "Lighting Source Locations from VHF Radiation Data for a Flash at Kennedy Space Center," Journal of Geophysical Research, 85 (C9): 4893-4903 (September 20, 1980).

26. Strong, Ian B. "The Direction of the Cosmic Gamma-Ray Sources," Proceedings of Conference on Transient Cosmic Gamma- and X-Ray Sources, September 20-21, 1973. 10-15. Los Alamos Scientific Laboratory of the University of California, Los Alamos NM, 1973 (LA-5505-C).

27. Taylor, Angus E. Advanced Calculus. Boston: Ginn and Company, 1955.

28. Terrell, J. et al. "Observation of Two Gamma-Ray Bursts by VELA X-Ray Detectors," The Astrophysical Journal, 254: 279-286 (March 1, 1982).

29. Toman, K. and J. E. Martine. Study of Source Location Error by Computer Simulation, Air Force Cambridge Research Labs, L. G. Hanscom Field, Ma, November, 1973 (AFCRL-TR-73-0634, AD-775 069).

30. Turman, B. N. "Detection of Lighting Superbolts," Journal of Geophysical Research, 82 (18): 2566-2568 (June 20, 1977).

31. Uman, Martin A. and et. al. "An Unusual Lighting Flash at Kennedy Space Center," Science, 201 (4350): 9-16 (July 7, 1978).

32. Van Dierendonck, A. J. et al. "The Approach to Satellite Ephemeris Determination for the NAVSTAR Global Positioning System," Navigation Journal of the Institute of Navigation, 23: 76-86 (Spring, 1976).

33. Wood, Lawrence C. and Sven Treitel. "Seismic Signal Processing," Proceedings of the IEEE, 63 (4): 649-670 (April 1975)

VITA

Captain Chester M. Wozniakowski was born on 16 January 1951 in Ware, Massachusetts. He graduated from high school in Ware, Massachusetts, in 1968. He attended Providence College, Providence, Rhode Island, from which he received the degree of Bachelor of Arts in Mathematics in May 1972, and Lowell Technological Institute, Lowell, Massachusetts, from which he received the degree of Masters of Science in Mathematics in May 1974. He received a commission in the USAF through the OTS program in May 1977. His first assignment was at NORAD Combat Operations Center, Cheyenne Mountain Complex, Colorado Springs, Colorado as an Orbital Analyst. He then served as a Deputy Space System Director at the 12th Missile WArning Group, Thule AB, Greenland. His next assignment was at HQ SAC, Space surveillance and Missile Warning Directorate, Offutt AFB, Nebraska as a Missile Warning Requirements Officer until entering the School of Engineering, Air Force Institute of Technology, in May 1983.

Permanent address: 13 New Hampshire Ave
Three Rivers, Massachusetts 01080

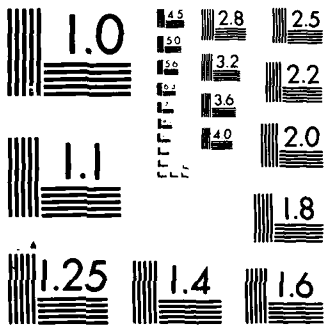
AD-A159 243 TIME DELAY OF ARRIVAL LOCATION ASSESSMENT USING FOUR SATELLITES(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING C M WOZNIAKOWSKI DEC 84 2/2

UNCLASSIFIED AFIT/GSO/PH/84D-5 F/G 17/7 NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

AD-A159 243

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: Unclassified

1b. RESTRICTIVE MARKINGS:

2a. SECURITY CLASSIFICATION AUTHORITY:

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE:

3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S): AFIT/GSO/PH/84D-5

5. MONITORING ORGANIZATION REPORT NUMBER(S):

6a. NAME OF PERFORMING ORGANIZATION: School of Engineering AF Institute of Technology

6b. OFFICE SYMBOL (If applicable): AFIT/EN

7a. NAME OF MONITORING ORGANIZATION:

6c. ADDRESS (City, State and ZIP Code): Wright-Patterson AFB, OH 45433

7b. ADDRESS (City, State and ZIP Code):

8a. NAME OF FUNDING/SPONSORING ORGANIZATION: AF Technical Applications Center

8b. OFFICE SYMBOL (If applicable): AFTAC/TXN

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER:

8c. ADDRESS (City, State and ZIP Code): Patrick AFB FL 32925

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.

11. TITLE (Include Security Classification): see Box 19

12. PERSONAL AUTHOR(S): Wozniakowski, Chester M., Capt, USAF

13a. TYPE OF REPORT: Thesis

13b. TIME COVERED: FROM ______ TO ______

14. DATE OF REPORT (Yr., Mo., Day): 1984, December

15. PAGE COUNT: 87

16. SUPPLEMENTARY NOTATION:

Approved for public release: IAW AFR 190-17.
LYNN E. WOLAVER 19 AUG 85
Dean for Research and Professional Development
Air Force Institute of Technology (ATC)
Wright-Patterson AFB OH 45433

17. COSATI CODES

FIELD	GROUP	SUB. GR.
17	08	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number): Time; Position; Optical Detection; Space Surveillance Systems; Navigation Satellites

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

11. (cont.) Title: Time Delay of Arrival Location Assessment Using Four Satellites
Thesis Chairman: James J. Lange, Maj, USAF

19. Abstract: The Time of Arrival (TOA)/ Time Delay of Arrival (TDOA) concept for locating a source are reviewed. The Vela satellite program and how they were used in conjunction with the TOA concept is discussed. NAVSTAR/GPS is reviewed next and how this interrelates with the other concepts discussed is mentioned. The last concept mentioned is that of nuclear detection. Again, examples on how this idea interrelates with the previous mentioned concepts are shown.

The problem addressed here is to develop mathematical formulas

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT: UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐

21. ABSTRACT SECURITY CLASSIFICATION: Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL: James J. Lange, Maj, USAF

22b. TELEPHONE NUMBER (Include Area Code): 513-255-4498

22c. OFFICE SYMBOL: AFIT/ENP

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE. UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

that describe the problem of having four satellites viewing an event, then accurately determining the location and time for that event. The first method uses a TOA technique to solve for the event's location and time. The second method employs the solution from the first method to predict uncertainty in location and time. Also, this uncertainty is determined through the use of implicit differentiation. These results are then compared with the projected difference from the standdard solution when the associated value is varied by an equivalent amount, in the first method. Results from one example are discussed.

END

FILMED

10-85

DTIC